

**FIRAT ÜNİVERSİTESİ**  
**Mühendislik Fakültesi**  
**Bilgisayar Mühendisliği Bölümü**

**BMÜ329 VERİ TABANI SİSTEMLERİ**  
**DÖNEM PROJESİ RAPORU**

**MÜZİK ARŞİVİ**

<b>Grup No</b>	03
<b>Takım Üyesi 1</b>	235260024-Şeyma Nur ŞAHİN
<b>Takım Üyesi 2</b>	235260030-Abdulkerim YILMAZ
<b>Takım Üyesi 3</b>	235260016-Ziya EKİZ
<b>Ders Sorumlusu</b>	Ertan BÜTÜN
<b>Teslim Tarihi</b>	27/12/2025

## İÇİNDEKİLER

1. **GİRİŞ VE PROJE TANIMI**
  - 1.1 Projenin Amacı ve Kapsamı
  - 1.2 Hedef Kullanıcılar
2. **PROJE GEREKSİNİMLERİ**
  - 2.1 Fonksiyonel Gereksinimler
  - 2.2 Fonksiyonel Olmayan Gereksinimler
  - 2.3 İş Kuralları ve Kısıtlamalar
3. **VERİ TABANI TASARIMI – VARLIK-İLİŞKİ (E-R) MODELİ**
  - 3.1 E-R Diyagramı
4. **İLİŞKİSEL ŞEMALAR**
  - 4.1 E-R'dan İlişkisel Şemalara Dönüşüm
5. **NORMALİZASYON**
  - 5.1 Normalizasyon Süreci
  - 5.2 Son Normalize Edilmiş Şema
6. **SQL SERVER VERİ TABANI ŞEMASI**
  - 6.1 Tablo Oluşturma Komutları
  - 6.2 Veri Tabanı Diyagramı
7. **ÖRNEK VERİLER**
  - 7.1 Veri Ekleme Komutları (INSERT)
8. **SQL KOMUTLARI VE SCRIPT DOSYALARI**
  - 8.1 Gereksinim Bazlı SQL Komutları
9. **SAKLI YORDAM VE TETİKLEYİCİ**
  - 9.1 Saklı Yordam (Stored Procedure)
    - 9.1.1 Amaç ve İş Operasyonu
    - 9.1.2 Saklı Yordam Kodu
    - 9.1.3 Test Senaryoları ve Sonuçları
  - 9.2 Tetikleyici (Trigger)
10. **TRANSACTION YÖNETİMİ**
  - 10.1 Transaction Senaryosu
  - 10.2 Transaction Kodu
  - 10.3 Başarılı Senaryo Testi
  - 10.4 Hata Senaryosu ve ROLLBACK Testi

## 11. TAKIM ÇALIŞMASI VE GÖREV DAĞILIMI

11.1 Takım Üyeleri ve Roller

11.2 Gerçekleştirilen İşler ve Sorumluluk Matrisi

## 12. SONUÇ VE DEĞERLENDİRME

# 1. GİRİŞ VE PROJE TANIMI

## 1.1 Projenin Amacı ve Kapsamı

Bu proje, kullanıcıların müzik verilerini (şarkılar, albümler, sanatçılar ve türler) düzenli bir şekilde yönetmesini sağlayan bir **Müzik Arşiv Sistemi** geliştirmeyi amaçlamaktadır.

Sistem, kullanıcıların kayıt olup kendi çalma listelerini oluşturabilecekleri, yöneticilerin ise sistemdeki verileri yönetebileceği bir yapıya sahiptir.

Proje, hem **kullanıcılar (dinleyiciler)** hem de **sanatçı** için farklı yetki seviyeleri tanımlar.

## 1.2 Hedef Kullanıcılar

### Kullanıcılar:

Sisteme kayıt olabilir, giriş yapabilir, şarkıları albüm, sanatçı veya tür bazlı arayabilir, çalma listeleri oluşturabilir ve bu listelere şarkı ekleyip çıkarabilir.

### Sanatçılar:

Sistemde sanatçı rolüne sahip kullanıcılar albüm, şarkı ve tür bilgileri ekleyebilir, mevcut kayıtları güncelleyebilir veya silebilir.

## 2. PROJE GEREKSİNİMLERİ

### 2.1 Fonksiyonel Gereksinimler

ID	Gereksinim Adı	Açıklama
FR-01	Kullanıcı Girişi	Sisteme kullanıcı adı ve şifre ile giriş yapar.
FR-02	Sanatçı Ekleme	Yetkili kullanıcılar sisteme yeni sanatçı bilgileri ekler.
FR-03	Albüm Ekleme	Seçilen bir sanatçıya ait albüm bilgileri sisteme ekler.
FR-04	Şarkı Ekleme	Albüm ve müzik türü seçilerek şarkı bilgileri sisteme kayder.
FR-05	Müzik Türü Yönetimi	Sistem yöneticisi müzik türlerini ekler ve günceller.
FR-06	Şarkı Listeleme	Kullanıcılar şarkıları sanatçıya, albüme veya türe göre listeler.
FR-07	Arama	Şarkı adı, sanatçı adı veya albüm adına göre arama yapar.
FR-08	Kayıt Güncelleme	Sanatçı, albüm veya şarkı bilgileri günceller.
FR-09	Kayıt Silme	Yetkili kullanıcılar sanatçı, albüm veya şarkı kayıtlarını siler.
FR-10	Detay Görüntüleme	Bir şarkıya ait süre, yayın tarihi, albüm ve tür bilgileri görüntüler.

## 2.2 Fonksiyonel Olmayan Gereksinimler

ID	Gereksinim Adı	Açıklama
NFR-01	Güvenlik	Kullanıcı ve sanatçı idler şifrelenmiş olarak saklanır
NFR-02	Veri Bütünlüğü	Foreign key, primary key bütünlükleri sağlanmalı
NFR-03	Performans	Şarkı arama sorguları hızlı ve optimize çalışmalı
NFR-04	Yedekleme	Eklenen şarkılar ve albümlerin yedeklenebilir bir şekilde tutulması.
NFR-05	Erişebilirlik	Kullanıcıların şarkıya kolayca erişebilmesi ve sanatçının kolayca şarkı ekleyebilmesi.

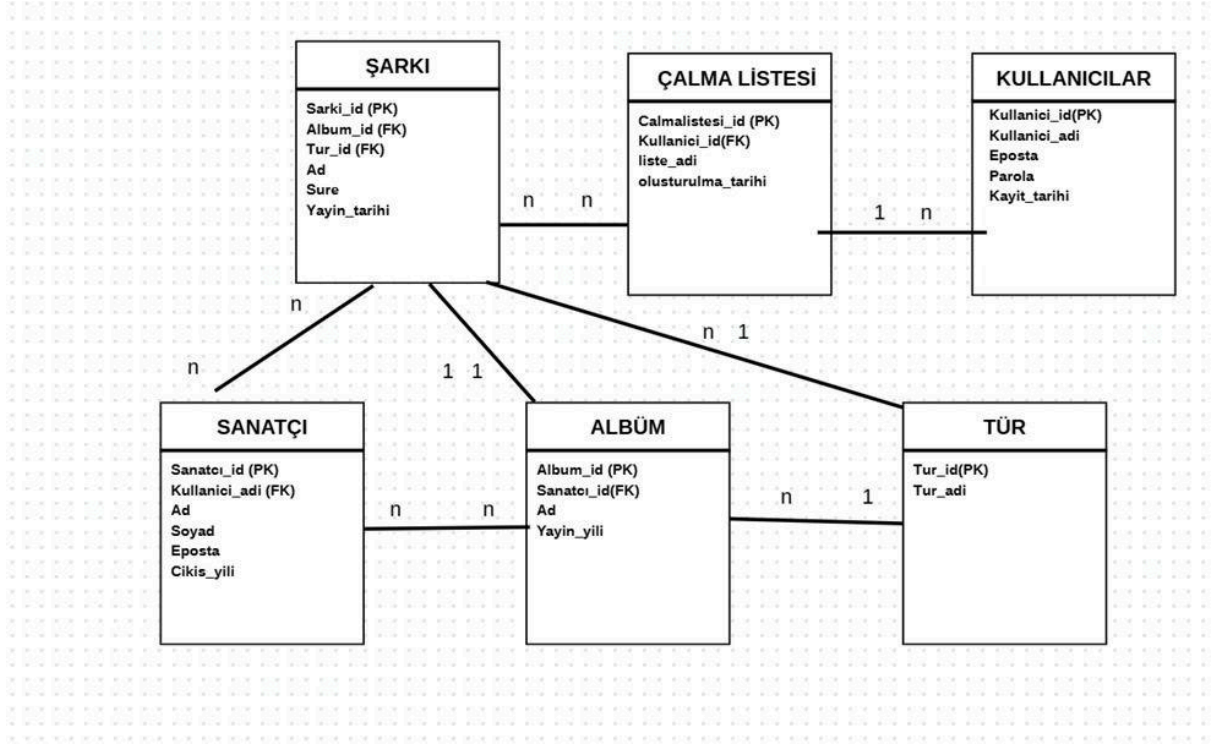
## 2.2 İş Kuralları, Kısıtlamalar

ID	Kısıtlamalar	Açıklama
01	Sanatçılar silme kısıtı	Sistemde bulunan hiçbir sanatçı kaydı silinemez. Veri bütünlüğünü korumak amacıyla sanatçı silme işlemleri engellenmiştir.
02	Sanatçı E-posta benzersizliği	Sanatçıların e-posta adresleri sistem genelinde benzersiz olmalıdır. Aynı e-posta adresi ile birden fazla sanatçı kaydı oluşturulamaz.
03	Çalma listesi otomatik silme	Kullanıcının çalma listesindeki son şarkı silindiğinde, ilgili çalma listesi otomatik olarak sistemden silinir.

ID	Kısıtlamalar	Açıklama
04	Kullanıcı E-posta format kontrolü	Kullanıcı e-posta adreslerinin geçerli bir e-posta formatında olması zorunludur. Geçersiz e-posta adresleri ile kullanıcı oluşturulamaz.
05	Tek hesap kuralları	Her kullanıcı bir e-posta dan sadece 1 hesap oluşturabilir. Daha önce kullanılmış bir e-posta adresi ile yeni hesap oluşturulmasına izin verilmez.
06	Kullanıcı adı benzersizliği	Kullanıcı adları sistem genelinde benzersiz olmalıdır. Aynı kullanıcı adına sahip birden fazla hesap oluşturulamaz.
07	Çalma Listesi Oluşturulma Tarihi Değiştirilemez	Çalma listesinin oluşturulma tarihi, liste oluşturulduktan sonra güncellenemez. Bu alan yalnızca ilk oluşturulma anını temsil eder.
08	Albüm İçinde Şarkı Adı Benzersizliği	Aynı albüm içerisinde aynı isimde birden fazla şarkı bulunamaz. Şarkı ekleme sırasında bu durum kontrol edilir.
09	Zorunlu Albüm ve Tür ilişkisi	Bir şarkı mutlaka bir albüme ve bir müzik türüne ait olmalıdır. Albüm veya tür bilgisi olmayan şarkı kaydı oluşturulamaz.

### 3. VERİ TABANI TASARIMI - VARLIK-İLİŞKİ (E-R) MODELİ

#### 3.1 E-R Diyagramı



## 4. İLİŞKİSEL ŞEMALAR

### 4.1 E-R'dan İlişkisel Şemalarına Dönüşüm

. ]ŞARKI - ÇALMA LİSTESİ  $n-n$

*SANATÇI-ŞARKI  $n-n$*

*ŞARKI -ALBÜM 1-1*

*ŞARKI-TÜRLER  $n-1$*

*SANATÇI - ALBÜM  $n-n$*

*ALBÜM- TÜRLER  $n-1$*

*ÇALMA LİSTESİ- KULLANICI 1-n*

**VARLIK ADI:**

**NİTELİKLER:**

**KULLANICILAR**      *Kullanici\_id(PK),Kullanici\_adi,Eposta,Parola,Kayit\_tarihi*

**SANATÇILAR**      *Sanatci\_id (PK),Kullanici\_adi ,Ad ,Soyad,Eposta,Cikis\_yili*

**ALBÜMLER**              *Album\_id (PK),Sanatci\_id(FK),Ad ,Yayin\_yili*

**TÜRLER**                      *Tur\_id(PK),Tur\_adi*

**ŞARKILAR**                      *Sarki\_id (PK),Album\_id (FK),Tur\_id (FK),Ad,Sure,Yayin\_tarihi*

**ÇALMA\_LİSTELERİ**                      *Calmalistes\_i\_id (PK),Kullanici\_id(FK),liste\_adi,olusturulma\_tarihi*



## 5. NORMALİZASYON

### 5.1 Normalizasyon Süreci

Normalizasyon işlemleri **1NF**, **2NF** ve **3NF** adımları izlenerek gerçekleştirilmiştir. Mevcut yapı **BCNF koşullarını tam olarak sağlamadığından**, tasarım **3NF** seviyesinde bırakılmıştır.

3NF olduğundan aslında hem 1NF hem de 2NF'i kapsar bundan ötürü hepsini adım adım açıkladık.

#### 1NF

- Tek bir hücrede birden fazla veri bulunmuyor.
- Tekrarlayan grup veya çok değerli nitelik yok.
- Varlıklar ayrı tablolara ayrılmıştır.

-SARKILAR  
-ALBUMLER  
-SANATCILAR  
-TURLER

1NF sağlanmıştır.

#### 2NF

Birincil anahtar bileşik ve anahtar olmayan tüm alanlar anahtarın tamamına bağımlıdır.

Çalma\_Listesi – Şarkı ilişkisi en önemli örneğidir.

CALMA\_LISTESI\_SARKI

(Calmalistesi\_id, Sarki\_id)

Başka hiçbir alan bulunmadığı için kısmi bağımlılık oluşmamaktadır ve 2NF dir.

#### 3NF

- Anahtar olmayan alanlar arasında geçişli bağımlılık yoktur
- Tür bilgileri (Tur\_adi) SARKILAR tablosunda tutulmamış
- Ayrı bir TURLER tablosu oluşturulmuş

- SARKILAR tablosu yalnızca Tur\_id (FK) içermiştir
- Sanatçı bilgileri ALBUMLER veya SARKILAR içinde tekrar edilmemiş
- SANATCILAR tablosu ayrı tutulmuştur
- Güncelleme anomalileri önlenmiş
- Veri tekrarı azaltılmış
- Geçişli bağımlılıklar ortadan kaldırılmıştır

bu maddeler sağlandığından 3NF dir.

### BCNF

BCNF değildir çünkü BCNF için her belirleyicinin süper anahtar olması gerekmektedir. Ancak:

- Sanatçı–Albüm ve Sanatçı–Şarkı ilişkilerinde
- İş kurallarına bağlı bağımlılıklar bulunduğundan

Tasarım **BCNF yerine 3NF** seviyesinde bırakıldı.

## 5.2 Son Normalize Edilmiş Şema

### KULLANICILAR

(Kullanici\_id **PK**, Kullanici\_adi, Eposta, Parola, Kayit\_tarihi)

### SANATCILAR

(Sanatci\_id **PK**, Kullanici\_adi, Ad, Soyad, Eposta, Cikis\_yili)

### ALBUMLER

(Album\_id **PK**, Sanatci\_id **FK**, Ad, Yayin\_yili)

### TURLER

(Tur\_id **PK**, Tur\_adi)

### **SARKILAR**

(Sarki\_id **PK**, Album\_id **FK**, Tur\_id **FK**, Ad, Sure, Yayin\_tarihi)

### **CALMA\_LISTELERI**

(Calmalistesi\_id **PK**, Kullanici\_id **FK**, Liste\_adi, Olusturulma\_tarihi)

### **CALMA\_LISTESI\_SARKI**

(Calmalistesi\_id **FK**, Sarki\_id **FK**)

## **6. SQL SERVER VERİ TABANI ŞEMASI**

### **6.1 Tablo Oluşturma Komutları**

```
CREATE DATABASE MuzikDB;
```

```
GO
```

```
USE MuzikDB;
```

```
GO
```

```
CREATE TABLE Kullanicilar (
```

```
    Kullanici_id INT IDENTITY(1,1) PRIMARY KEY,
```

```
    Kullanici_adi NVARCHAR(50) NOT NULL,
```

```
    Eposta NVARCHAR(100) NOT NULL UNIQUE,
```

```
    Parola NVARCHAR(255) NOT NULL,
```

```
    Kayit_tarihi DATETIME2 NOT NULL DEFAULT SYSUTCDATETIME()
```

```
);
```

```
GO
```

```
CREATE TABLE Sanatcilar (
```

```
    Sanatci_id INT IDENTITY(1,1) PRIMARY KEY,
```

```
    Kullanici_adi NVARCHAR(50) NOT NULL,
```

```
    Ad NVARCHAR(50) NOT NULL,
```

```
    Soyad NVARCHAR(50) NULL,
```

```
    Eposta NVARCHAR(100) NOT NULL UNIQUE,
```

```
    Cikis_yili SMALLINT NULL,
```

```
    CONSTRAINT CK_Sanatcilar_CikisYili CHECK (Cikis_yili IS NULL OR (Cikis_yili  
BETWEEN 1900 AND 2100))
```

```
);  
GO  
CREATE TABLE Turler (  
    Tur_id INT IDENTITY(1,1) PRIMARY KEY,  
    Tur_adi NVARCHAR(50) NOT NULL UNIQUE  
);  
GO  
CREATE TABLE Albumler (  
    Album_id INT IDENTITY(1,1) PRIMARY KEY,  
    Sanatci_id INT NOT NULL,  
    Ad NVARCHAR(100) NOT NULL,  
    Yayin_yili SMALLINT NULL,  
    CONSTRAINT CK_Albumler_YayinYili CHECK (Yayin_yili IS NULL OR (Yayin_yili  
BETWEEN 1900 AND 2100)),  
    CONSTRAINT FK_Albumler_Sanatcilar  
        FOREIGN KEY (Sanatci_id) REFERENCES Sanatcilar(Sanatci_id)  
);  
GO  
CREATE TABLE Sarkilar (  
    Sarki_id INT IDENTITY(1,1) PRIMARY KEY,  
    Album_id INT NOT NULL,  
    Tur_id INT NOT NULL,  
    Ad NVARCHAR(150) NOT NULL,  
    Sure_saniye INT NOT NULL,  
    Yayin_tarihi DATE NULL,  
  
    CONSTRAINT CK_Sarkilar_Sure CHECK (Sure_saniye > 0 AND Sure_saniye <=  
60*60*2),  
  
    CONSTRAINT FK_Sarkilar_Albumler  
        FOREIGN KEY (Album_id) REFERENCES Albumler(Album_id),  
  
    CONSTRAINT FK_Sarkilar_Turler
```

```
        FOREIGN KEY (Tur_id) REFERENCES Turler(Tur_id)
    );
GO
CREATE TABLE CalmaListeleri (
    Calmalistesi_id INT IDENTITY(1,1) PRIMARY KEY,
    Kullanici_id INT NOT NULL,
    Liste_adi NVARCHAR(100) NOT NULL,
    Olusturulma_tarihi DATETIME2 NOT NULL DEFAULT SYSUTCDATETIME(),

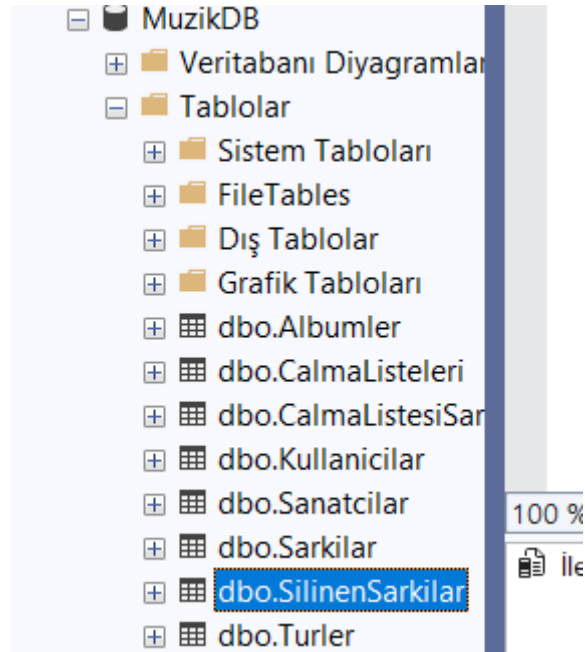
    CONSTRAINT FK_CalmaListeleri_Kullanici
        FOREIGN KEY (Kullanici_id) REFERENCES Kullanici(Kullanici_id)
    );
GO
CREATE TABLE CalmaListesiSarkilari (
    Calmalistesi_id INT NOT NULL,
    Sarki_id INT NOT NULL,
    Eklenme_tarihi DATETIME2 NOT NULL DEFAULT SYSUTCDATETIME(),
    Sira INT NULL,

    CONSTRAINT PK_CalmaListesiSarkilari PRIMARY KEY (Calmalistesi_id,
    Sarki_id),

    CONSTRAINT FK_CLS_CalmaListeleri
        FOREIGN KEY (Calmalistesi_id) REFERENCES
        CalmaListeleri(Calmalistesi_id)
        ON DELETE CASCADE,

    CONSTRAINT FK_CLS_Sarkilar
        FOREIGN KEY (Sarki_id) REFERENCES Sarkilar(Sarki_id)
    );
GO
```

### 6.3 Veri Tabanı Diyagramı



## 7. ÖRNEK VERİLER

### 7.1 Veri Ekleme Komutları (INSERT)

USE MuzikDB;

GO

INSERT INTO Turler (Tur\_adi)

VALUES

--ilk önce 8 tane tür ekledik ve bunların idleri aklımızda kalsın diye yanda yazık

('Pop'), -- ID:1

('Rock'), -- ID: 2

('Rap'), -- ID: 3

('Metal'), -- ID:4

('R&B'), -- ID: 5

('Arabesk'), -- ID 6

('Alternatif'), -- ID: 7

('Indie'); -- ID 8

INSERT INTO Kullanicilar (Kullanici\_adi, Eposta, Parola)

VALUES

--Burda sırasıyla kullanıcı adlarını epostaları ve parolarını aldık ve bunların gerçekçi olmasını istedik

('ahmet8', 'ahmet@gmail.com', '1234') ,

('zeynep7', 'zeynep@gmail.com', '1234') ,

('kazım6', 'kazım@gmail.com', '1234') ,

('dila5', 'dila@gmail.com', '1234') ,

('burakemre4', 'burak@gmail.com', '1234') ,

```
('dilara3', 'dilara@gmail.com', '1234') ,  
( 'furkan2', 'furkan@gmail.com', '1234') ,  
( 'aysegul1', 'aysegul@gmail.com', '1234' ),  
( 'ziya123', 'ziya@gmail.com', '1234' ) ,  
( 'sheyma', 'sheyma@gmail.com', '1234')  
;
```

```
INSERT INTO Sanatcilar (Kullanici_adi, Ad, Soyad, Eposta, Cikis_yili)  
VALUES
```

--sanatçıların gerçek sanatçılardan bilgilerini aldık ama epostalarını kendimiz atadık  
ve çıkış tarihleride girdik

```
('sezen_Aksu', 'Sezen', 'Aksu', 'sezen@aksu.com', 1975 ),  
( 'muslum_baba', 'Müslüm', 'Gürses', 'baba@muslum.com', 1978),  
( 'tarkan', 'Tarkan', 'Tevetoğlu', 'tarkan@megastar.com', 1992 ),  
( 'duman', 'Duman', NULL, 'info@duman.com', 1999),  
( 'ceza', 'Bilgin', 'Özçalkan', 'ceza@rap.com', 2002) ,  
( 'manga', 'maNga', NULL, 'iletisim@manga.com', 2001) ,  
( 'sebnem_ferah', 'Şebnem', 'Ferah', 'sebo@rock.com', 1996),  
( 'eminem', 'Marshall', 'Mathers', 'eminem@shady.com', 1996) ,  
( 'rihanna', 'Rihanna', 'Fenty', 'riri@fenty.com', 2005),  
( 'metallica', 'Metallica', NULL, 'band@metallica.com', 1981),  
( 'queen', 'Queen', NULL, 'freddie@queen.com', 1970) ,  
( 'adele', 'Adele', 'Adkins', 'adele@uk.com', 2008),  
( 'weeknd', 'Abel', 'Teschfaye', 'weeknd@xo.com', 2010),  
( 'lp_band', 'Linkin Park', NULL, 'lp@nu.com', 2000),  
( 'bruno_mars', 'Bruno', 'Mars', 'bruno@mars.com', 2010);
```



INSERT INTO Albumler (Sanatci\_id, Ad, Yayin\_yili)

VALUES

--burda sırasıyla santçılardan aldığımız bazı albimleri girdik

- (1, 'Gülümse', 1991),
- (1, 'Biraz Pop Biraz Sezen', 2017),
- (2, 'Sandık', 2009),
- (2, 'Aşk Tesadüfleri Sever', 2006),
- (3, 'Karma', 2001),
- (3, '10', 2017),
- (4, 'Belki Alışman Lazım', 2002),
- (4, 'Seni Kendime Sakladım', 2005),
- (5, 'Rapstar', 2004),
- (5, 'Yerli Plaka', 2006),
- (6, 'Şehr-i Hüzün', 2009),
- (6, 'maNga', 2004),
- (7, 'Kadın', 1996),
- (7, 'Od', 2013),
- (8, 'The Eminem Show', 2002),
- (8, 'Recovery', 2010),
- (9, 'Good Girl Gone Bad', 2007),
- (9, 'Anti', 2016),
- (10, 'Black Album', 1991),
- (10, 'Master of Puppets', 1986),
- (11, 'A Night at the Opera', 1975),
- (12, '21', 2011),
- (13, 'Starboy', 2016),
- (13, 'After Hours', 2020),
- (14, 'Hybrid Theory', 2000),
- (14, 'Meteora', 2003),
- (15, 'Doo-Wops & Hooligans', 2010),
- (15, '24K Magic', 2016);

INSERT INTO Sarkilar (Album\_id, Tur\_id, Ad, Sure\_saniye, Yayin\_tarihi) VALUES

-- burda albumleri girdik ve şarkıları girdik ve sürelerini ve çıkış tarihlerini girdik  
--ama ilk başta arkıların bulundukları albümleri ve daha sonrasında türlerini daha sonrasinde geri kalan verileri girdik

(1, 1, 'Gülümse', 285, '1991-01-01'),  
(1, 1, 'Vazgeçtim', 309, '1991-01-01'),  
(1, 1, 'Hadi Bakalım', 296, '1991-01-01'),  
(2, 1, 'İhanetten Geri Kalan', 250, '2017-05-15'),  
(3, 6, 'İtirazım Var', 245, '2009-02-01'),  
(3, 6, 'Tutamıyorum Zamanı', 230, '2009-02-01'),  
(4, 7, 'Nilüfer', 260, '2006-04-18'),  
(4, 7, 'Affet', 285, '2006-04-18'),  
(5, 1, 'Kuzu Kuzu', 232, '2001-07-15'),  
(5, 1, 'Hüp', 218, '2001-07-15'),  
(5, 1, 'Verme', 290, '2001-07-15'),  
(6, 1, 'Yolla', 226, '2017-06-15'),  
(7, 2, 'Bu Akşam', 205, '2002-04-04'),  
(7, 2, 'Her Şeyi Yak', 256, '2002-04-04'),  
(8, 2, 'Seni Kendime Sakladım', 242, '2005-06-07'),  
(8, 2, 'Aman Aman', 235, '2005-06-07'),  
(9, 3, 'Holocaust', 208, '2004-01-01'),  
(9, 3, 'Neyim Var Ki', 245, '2004-01-01'),  
(10, 3, 'Yerli Plaka', 230, '2006-09-01'),  
(10, 3, 'Fark Var', 215, '2006-09-01'),  
(11, 7, 'Cevapsız Sorular', 275, '2009-04-15'),  
(11, 7, 'Dünyanın Sonuna Doğmuşum', 260, '2009-04-15'),

(12, 7, 'Bir Kadın Çizeceksin', 238, '2004-12-14'),  
(13, 2, 'Vazgeçtim Dünyadan', 285, '1996-11-15'),  
(13, 2, 'Yağmurlar', 295, '1996-11-15'),  
(15, 3, 'Without Me', 290, '2002-05-26'),  
(15, 3, 'Sing for the Moment', 339, '2002-05-26'),  
(15, 3, 'Cleanin Out My Closet', 297, '2002-05-26'),  
(16, 3, 'Not Afraid', 248, '2010-04-29'),  
(16, 3, 'Love The Way You Lie', 263, '2010-04-29'),  
(17, 5, 'Umbrella', 275, '2007-03-29'),  
(17, 5, 'Shut Up and Drive', 213, '2007-03-29'),  
(18, 5, 'Work', 219, '2016-01-27'),  
(19, 4, 'Enter Sandman', 331, '1991-08-12'),  
(19, 4, 'The Unforgiven', 387, '1991-08-12'),  
(19, 4, 'Nothing Else Matters', 388, '1991-08-12'),  
(20, 4, 'Master of Puppets', 515, '1986-03-03'),  
(21, 2, 'Bohemian Rhapsody', 354, '1975-10-31'),  
(21, 2, 'Love of My Life', 217, '1975-10-31'),  
(22, 1, 'Rolling in the Deep', 228, '2011-01-24'),  
(22, 1, 'Someone Like You', 285, '2011-01-24'),  
(22, 1, 'Set Fire to the Rain', 242, '2011-01-24'),  
(23, 5, 'Starboy', 230, '2016-09-21'),  
(24, 5, 'Blinding Lights', 200, '2019-11-29'),  
(24, 5, 'Save Your Tears', 215, '2020-03-20'),  
(25, 2, 'In the End', 216, '2000-10-24'),  
(25, 2, 'Crawling', 209, '2000-10-24'),  
(26, 2, 'Numb', 187, '2003-03-25'),  
(27, 1, 'Grenade', 222, '2010-09-28'),  
(27, 1, 'Just the Way You Are', 220, '2010-09-28'),  
(28, 1, '24K Magic', 226, '2016-10-07');

INSERT INTO CalmaListeleri (Kullanici\_id, Liste\_adi)

VALUES

--burda her kullanıcının kendine özgü olan çalma listelerini girdik bu çalma listeleri n-n şeklide yaptık çuku her kullanıcı birden fazla çalma listeleri olabilir birde

--herkese açık çalma listeleri olaiblr

(1, 'Ahmet Favori'),

(1, 'piyasa Şarkıları'),

(3, 'Sadece Rock'),

(3, 'ağır metal'),

(2, 'Türkçe'),

(7, 'Efkarlıyken'),

(10, 'Ders Çalışırken');

INSERT INTO CalmaListesiSarkilari (Calmalistesi\_id, Sarki\_id, Sira)

VALUES

--burda çalmalistelerine bazı şarkıları ekledik ve bu eklemeleri şarkıların idlerine göre yaptık ve sıralarını koduk

(1, 15, 1), (1, 19, 2),(1, 35, 3),

(2, 5, 1), (2, 25, 3), (2, 1, 2),

(3, 9, 1),(3, 10, 2),(3, 39, 3),(3, 41, 4),

(4, 19, 1),(4, 20, 2),(4, 22, 3),

(5, 30, 1),(5, 45, 2), (5, 7, 3),

(6, 5, 1),(6, 6, 2),(6, 2, 3);

GO

DESKTOP-JNG29IG\SQLEXP

Veritabanları

- Sistem Veritabanları
- Veritabanı Anlık Görün
- BANK
- BikeStores
- DENEME
- MuzikDB
  - Veritabanı Diyagran
  - Tablolar
    - Sistem Tabloları
    - FileTables
    - Dış Tablolar
    - Grafik Tabloları
    - dbo.Albumler
    - dbo.CalmaListele
    - dbo.CalmaListesi
    - dbo.Kullanıcılar
    - dbo.Sanatçılar
    - dbo.Sarkılar
    - Sütunlar
      - Sarki\_id (Pk)
      - Album\_id (
      - Tur\_id (YA
      - Ad (nvarchar
      - Sure\_saniye
      - Yayin\_tarihi
    - Anahtarlar

SELECT \*

From Sanatçılar

100 %

1 0

Sonuçlar İletiler

	Sanatci_id	Kullanici_adi	Ad	Soyad	Eposta	Cikis_yili
1	1	sezen_Aksu	Sezen	Aksu	sezen@aksu.com	1975
2	2	muslum_baba	Müslüm	Gürses	baba@muslum.com	1978
3	3	tarkan	Tarkan	Tevetoğlu	tarkan@megastar.com	1992
4	4	duman	Duman	NULL	info@duman.com	1999
5	5	ceza	Bilgin	Özçalkan	ceza@rap.com	2002
6	6	manga	maNga	NULL	iletisim@manga.com	2001
7	7	sebnem_ferah	Şebnem	Ferah	sebo@rock.com	1996
8	8	eminem	Marshall	Mathers	eminem@shady.com	1996
9	9	rihanna	Rihanna	Fenty	riri@fenty.com	2005
10	10	metallica	Metallica	NULL	band@metallica.com	1981
11	11	queen	Queen	NULL	freddie@queen.com	1970
12	12	adele	Adele	Adkins	adele@uk.com	2008
13	13	weeknd	Abel	Tesfaye	weeknd@xo.com	2010
14	14	lp_band	Linkin Park	NULL	lp@nu.com	2000
15	15	bruno_mars	Bruno	Mars	bruno@mars.com	2010

Nesne Gezini

DESKTOP-JNG29IG\SQLEXP

Veritabanları

- Sistem Veritabanları
- Veritabanı Anlık Görün
- BANK
- BikeStores
- DENEME
- MuzikDB
  - Veritabanı Diyagran
  - Tablolar
    - Sistem Tabloları
    - FileTables
    - Dış Tablolar
    - Grafik Tabloları
    - dbo.Albumler
    - dbo.CalmaListele
    - dbo.CalmaListesi
    - dbo.Kullanıcılar
    - dbo.Sanatçılar

SQLQuery1.s...G\Shey (54))\*

SELECT \*

From Kullanicilar

100 %

1 0

Sonuçlar İletiler

	Kullanici_id	Kullanici_adi	Eposta	Parola	Kayit_tarihi
1	1	ahmet8	ahmet@gmail.com	1234	2025-12-23 10:01:00.0476300
2	2	zeynep7	zeynep@gmail.com	1234	2025-12-23 10:01:00.0476300
3	3	kazim6	kazim@gmail.com	1234	2025-12-23 10:01:00.0476300
4	4	dila5	dila@gmail.com	1234	2025-12-23 10:01:00.0476300
5	5	burakemre4	burak@gmail.com	1234	2025-12-23 10:01:00.0476300
6	6	dilara3	dilara@gmail.com	1234	2025-12-23 10:01:00.0476300
7	7	furkan2	furkan@gmail.com	1234	2025-12-23 10:01:00.0476300
8	8	aysegul1	aysegul@gmail.com	1234	2025-12-23 10:01:00.0476300
9	9	ziya123	ziya@gmail.com	1234	2025-12-23 10:01:00.0476300
10	10	sheyma	sheyma@gmail.com	1234	2025-12-23 10:01:00.0476300

Nesne Gezişini

SQLQuery1.s...G:\Shey (54)\*

1 SELECT  
2 From Sarkilar

100 % 1 0

Sat: 1 Kirt: 9 SEKMELER CRLF

Sarkil_id	Album_id	Tur_id	Ad	Sure_saniye	Yayin_tarihi
1	1	1	Gölümse	285	1991-01-01
2	2	1	Vazgeçtim	309	1991-01-01
3	3	1	Hadi Bakalım	296	1991-01-01
4	4	2	Harvestin Geli Kalan	250	2017-05-15
5	5	3	Brazim Var	245	2009-02-01
6	6	3	Tutamiyorum Zamanı	230	2009-02-01
7	7	4	Niçiler	260	2006-04-18
8	8	4	Affet	285	2006-04-18
9	9	5	Kuzu Kuzu	232	2001-07-15
10	10	5	Hop	218	2001-07-15
11	11	5	Verme	290	2001-07-15
12	12	6	Yolla	226	2017-06-15
13	13	7	Bu Akşam	205	2002-04-04
14	14	7	Her Şey Yak	295	2002-04-04
15	15	8	Seni Kendime Sakl...	242	2005-06-07
16	16	8	Aman Aman	235	2005-06-07
17	17	9	Holocaust	208	2004-01-01
18	18	9	Neyim Yar Ki	245	2004-01-01
19	19	10	Yeni Plaka	230	2006-09-01
20	20	10	Fark Var	215	2006-09-01
21	21	11	Cevapsız Sorular	275	2009-04-15
22	22	11	Dünyanın Sonuna ...	260	2009-04-15
23	23	12	Bir Kadın Çıksakın	238	2004-12-14
24	24	13	Vazgeçtim Dönyeden	285	1996-11-15
25	25	13	Yadimurur	295	1996-11-15
26	26	15	Without Me	290	2002-05-26
27	27	15	Sing for the Moment	339	2002-05-26
28	28	15	Clemin Out My Clo...	297	2002-05-26
29	29	16	Not Afraid	248	2010-04-29
30	30	16	Love The Way You ...	263	2010-04-29
31	31	17	Umbrella	275	2007-03-29
32	32	17	Shut Up and Drive	213	2007-03-29
33	33	18	Work	219	2016-01-27
34	34	19	Enter Sandman	331	1991-08-12
35	35	19	The Unforgiven	387	1991-08-12

Sorgu başarıyla yürütüldü.

DESKTOP-ING29IG\SQLEXPRESS ... DESKTOP-ING29IG\Shey (54) MuzikDB 00:00:00 51 satır

SQLQuery1.s...G\Shey (54))\*

```
1 SELECT *
2 From CalmaListeleri
```

100 % 1 0

Sonuçlar İletiler

	Calmalistesi_id	Kullanici_id	Liste_adi	Olusturulma_tarihi
1	1	1	Ahmet Favori	2025-12-23 10:01:00.0486304
2	2	1	piyasa Şarkıları	2025-12-23 10:01:00.0486304
3	3	3	Sadece Rock	2025-12-23 10:01:00.0486304
4	4	3	ağır metal	2025-12-23 10:01:00.0486304
5	5	2	Türkçe	2025-12-23 10:01:00.0486304
6	6	7	Efkarlıyken	2025-12-23 10:01:00.0486304
7	7	10	Ders Çalışırken	2025-12-23 10:01:00.0486304

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Server Explorer' pane displays the database structure, including tables like 'dbo.Albumler'. The central 'Query Window' contains the following SQL query:

```
1 SELECT *
2 FROM Albumler
```

Below the query window, the 'Results' pane shows a table with 5 columns: 'Album\_id', 'Sanatci\_id', 'Ad', and 'Yayin\_yili'. The table contains 28 rows of data, representing a list of albums and their artists.

	Album_id	Sanatci_id	Ad	Yayin_yili
1	1	1	Gülümse	1991
2	2	1	Biraz Pop Biraz Sezen	2017
3	3	2	Sandık	2009
4	4	2	Aşk Tesadüfleri Sever	2006
5	5	3	Karma	2001
6	6	3	10	2017
7	7	4	Belki Alışman Lazım	2002
8	8	4	Seni Kendime Sakladım	2005
9	9	5	Rapstar	2004
10	10	5	Yerli Plaka	2006
11	11	6	Şehr-i Hüzün	2009
12	12	6	maNga	2004
13	13	7	Kadın	1996
14	14	7	Od	2013
15	15	8	The Eminem Show	2002
16	16	8	Recovery	2010
17	17	9	Good Girl Gone Bad	2007
18	18	9	Anti	2016
19	19	10	Black Album	1991
20	20	10	Master of Puppets	1986
21	21	11	A Night at the Opera	1975
22	22	12	21	2011
23	23	13	Starboy	2016
24	24	13	After Hours	2020
25	25	14	Hybrid Theory	2000
26	26	14	Meteora	2003
27	27	15	Doo-Wops & Hooligans	2010
28	28	15	24K Magic	2016

## 8. SQL KOMUTLARI VE SCRIPT DOSYALARI

### 8.1 Gereksinim Bazlı SQL Komutları ve Script Dosyaları

**Gereksinim 1:** Sistemdeki tüm sanatçılar listelenebilmelidir

SQL Komutu:

```
SELECT Sanatci_id, Ad, Eposta
```

```
FROM Sanatcilar;
```

Bu sorgu, sistemde kayıtlı olan tüm sanatçıların temel bilgilerini listelemek için kullanılmıştır.



**Gereksinim 2:** Bir sanatçıya ait albümler listelenebilmelidir

SQL Komutu:

```
SELECT a.Album_id, a.Ad AS AlbumAdi, a.Yayin_tarihi  
FROM Albumler a  
INNER JOIN Sanatcilar s ON a.Sanatci_id = s.Sanatci_id  
WHERE s.Ad = 'Tarkan';
```

Bu komut Belirli bir sanatçıya ait albümlerin listelenmesini sağlar.

**Gereksinim 3:** Albüm bilgileriyle birlikte şarkılar listelenebilmelidir

SQL Komutu:

```
SELECT s.Ad AS SarkiAdi, a.Ad AS AlbumAdi, s.Sure_saniye  
FROM Sarkilar s  
INNER JOIN Albumler a ON s.Album_id = a.Album_id;
```

Şarkılar albüm bilgileriyle birlikte görüntülenmektedir.

**Gereksinim 4:** Şarkılar müzik türleriyle birlikte listelenebilmelidir

SQL Komutu:

```
SELECT s.Ad AS SarkiAdi, t.Ad AS TurAdi  
FROM Sarkilar s  
INNER JOIN Turler t ON s.Tur_id = t.Tur_id;
```

Şarkıların hangi müzik türüne ait olduğu bilgisiyle listelenmesini sağlar.

**Gereksinim 5:** Belirli bir müzik türüne ait şarkılar listelenebilmelidir

SQL Komutu:

```
SELECT s.Ad AS SarkiAdi  
FROM Sarkilar s  
INNER JOIN Turler t ON s.Tur_id = t.Tur_id  
WHERE t.Ad = 'Pop';
```

Seçilen müzik türüne ait şarkıların listelenmesi için kullanılmıştır.

**Gereksinim 6:** Şarkı ekleme işlemi yapılabilmelidir

SQL Komutu:

```
INSERT INTO Sarkilar (Album_id, Tur_id, Ad, Sure_saniye,  
Yayin_tarihi)  
VALUES (1, 1, 'Yeni Şarkı', 210, '2022-04-15');
```

Yeni bir şarkının sisteme eklenmesini sağlar. Bu işlem sırasında tetikleyiciler ve kısıtlamalar devreye girmektedir.

**Gereksinim 7:** Şarkı bilgileri güncellenebilmelidir

SQL Komutu:

```
UPDATE Sarkilar  
SET Sure_saniye = 230  
WHERE Sarki_id = 5;
```

Mevcut bir şarkının süresinin güncellenmesini sağlar.

**Gereksinim 8:** Şarkı silme işlemi yapılabilmelidir

SQL Komutu:

```
DELETE FROM Sarkilar  
WHERE Sarki_id = 10;
```

Belirtilen şarkı kaydını sistemden silmek için kullanılmıştır.

**Gereksinim 9:** Kullanıcı çalma listesine şarkı ekleyebilmelidir

SQL Komutu:

```
INSERT INTO CalmaListesiSarkilar (CalmaListesi_id, Sarki_id)  
VALUES (2, 4);
```

Kullanıcının çalma listesine şarkı eklemesini sağlar.

**Gereksinim 10:**

















Çalma listesindeki şarkılar listelenebilmelidir

SQL Komutu:

```
SELECT s.Ad AS SarkiAdi  
FROM CalmaListesiSarkilar cs  
INNER JOIN Sarkilar s ON cs.Sarki_id = s.Sarki_id  
WHERE cs.CalmaListesi_id = 2;
```

Belirli bir çalma listesinde bulunan şarkıları listeler.

**Script Dosyası:**

 MUZIKDB 15-16.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery1.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery3.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery4.sql	Microsoft SQL Server Quer...	3 KB	Hayır
 SQLQuery5.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery6.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery7.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery8.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery9.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery10.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery11.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery12.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery13.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery14.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery16.sql	Microsoft SQL Server Quer...	1 KB	Hayır
 SQLQuery17.sql	Microsoft SQL Server Quer...	1 KB	Hayır

## 9. SAKLI YORDAM VE TETİKLEYİCİ

### 9.1 Saklı Yordam Adı (Stored Procedure)

#### 9.1.1 Amaç ve İş Operasyonu

Bu saklı yordamın amacı, belirli bir albüme ait tüm şarkıları listelemektir.

Sistem içerisinde kullanıcıların veya sanatçıların albüm seçerek o albümde yer alan şarkıları görüntüleyebilmesi iş operasyonudur.

#### 9.1.2 Saklı Yordam Kodu

```
CREATE PROCEDURE sp_AlbumSarkilariniListele
```

```
    @AlbumId INT
```

```
AS
```

```
BEGIN
```

```

SELECT
    S.Sarki_id,
    S.Ad AS Sarki_Adi,
    S.Yayin_tarihi,
    T.Tur_adi
FROM Sarkilar S
INNER JOIN Turler T ON S.Tur_id = T.Tur_id
WHERE S.Album_id = @AlbumId;
END;

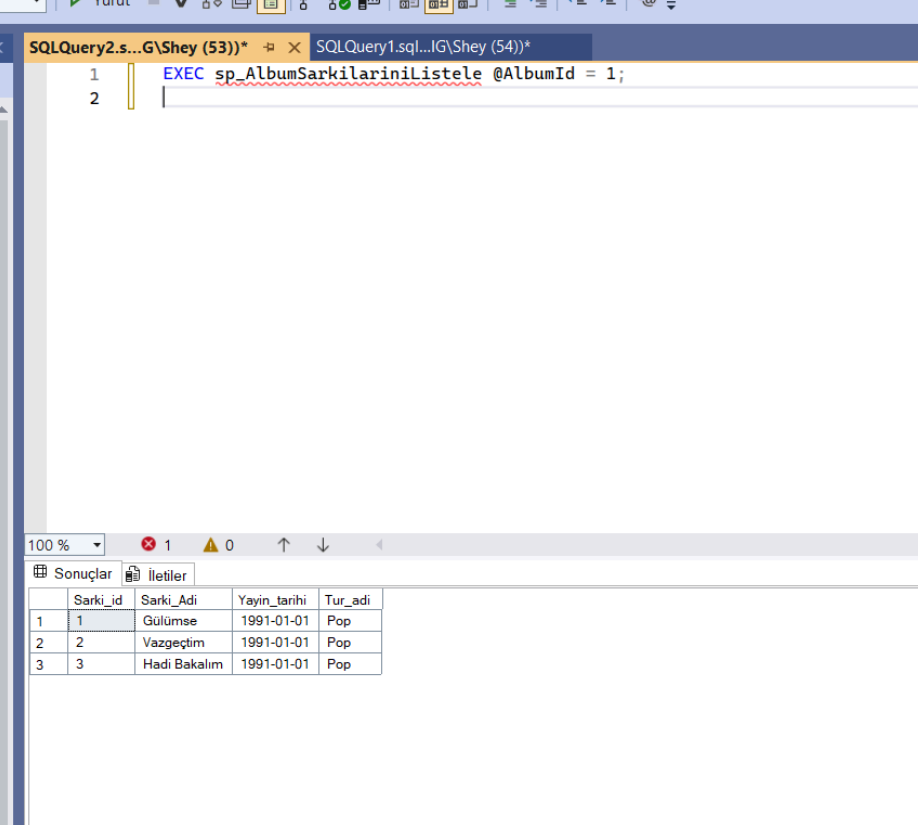
```

@AlbumId → Şarkıları listelenecek albümün ID bilgisini alır

INNER JOIN → Şarkıların tür bilgileri Turler tablosundan alınır

Sadece **ilgili albüme ait şarkılar** listelenir

### 9.1.3 Test Senaryoları ve Sonuçları



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a query window with the following SQL code:

```

1 EXEC sp_AlbumSarkilariniListele @AlbumId = 1;
2

```

The bottom pane shows the results of the query execution. The results are displayed in a table with the following columns: Sarki\_id, Sarki\_Adi, Yayin\_tarihi, and Tur\_adi. The table contains three rows of data:

	Sarki_id	Sarki_Adi	Yayin_tarihi	Tur_adi
1	1	Gülümse	1991-01-01	Pop
2	2	Vazgeçtim	1991-01-01	Pop
3	3	Hadi Bakalım	1991-01-01	Pop

## 9.2 Tetikleyici (Trigger)

### 9.2.1 Amaç ve Tetikleme Koşulu

#### Zorunlu Albüm ve Tür Kontrolü

**Amaç:**

Bir şarkının mutlaka bir albüme ve bir müzik türüne ait olmasını zorunlu kılarak eksik veri girişini engellemek.

**Tetikleme Koşulu:**

Sarkılar tablosu üzerinde **INSERT** veya **UPDATE** işlemi yapıldığında tetiklenir.

### 9.2.2 Tetikleyici Kodu

```
CREATE TRIGGER Sarki_ZorunluAlanVeReferansKontrolu
ON Sarkilar
AFTER INSERT, UPDATE
AS
BEGIN
```

```
-- (Tablo yapınızda NOT NULL olsa bile trigger mantığı gereği bu kontrolü buraya ekliyoruz)
```

```
IF EXISTS (SELECT * FROM inserted WHERE Album_id IS NULL OR Tur_id IS NULL)
```

```
BEGIN
```

```
    RAISERROR ('HATA: Bir şarkı mutlaka bir Albüme ve bir Müzik Türüne ait olmalıdır! (Alanlar boş bırakılamaz)', 16, 1);
```

```
    ROLLBACK TRANSACTION; -- İşlemi iptal et
```

```
    RETURN;
```

```
END
```

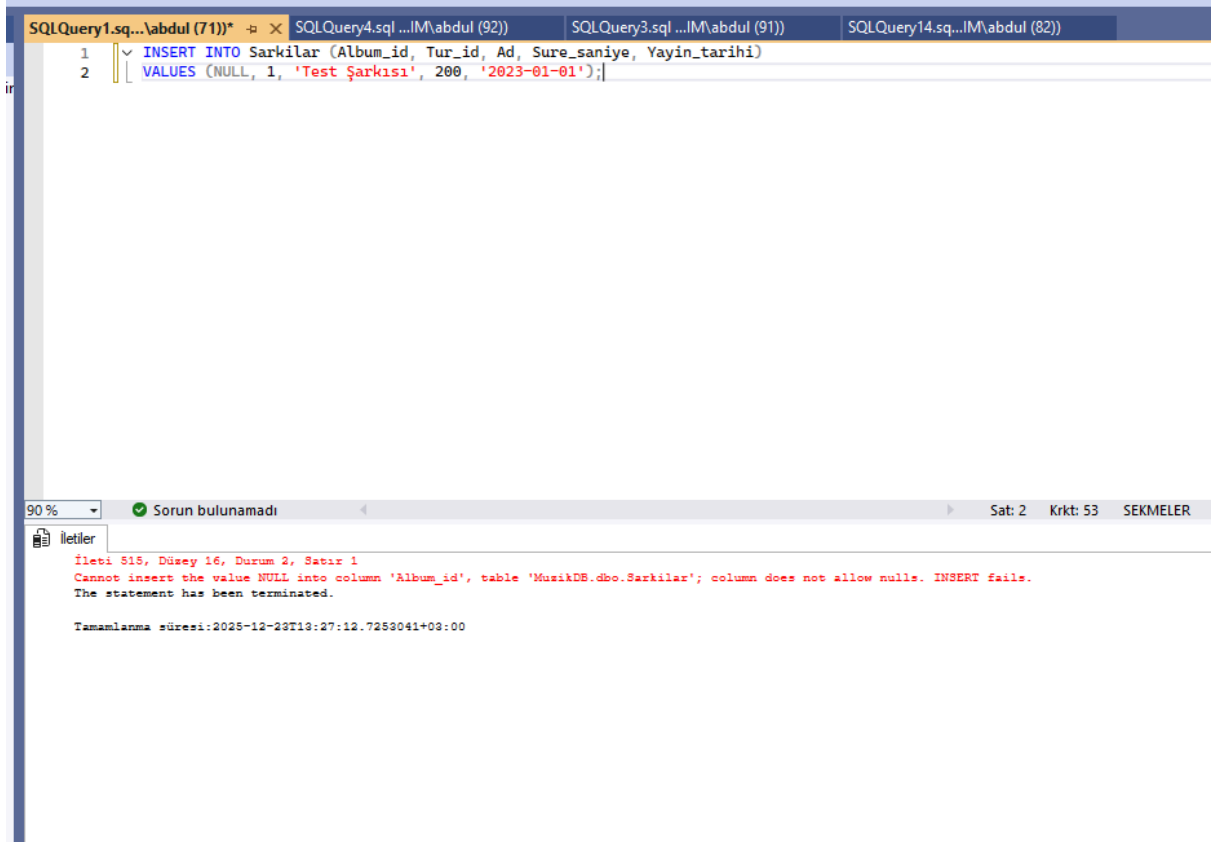
```
IF EXISTS (
```

```
SELECT i.Sarki_id
FROM inserted i
LEFT JOIN Albumler a ON i.Album_id = a.Album_id
WHERE a.Album_id IS NULL -- Eşleşen bir albüm bulunamadıysa
)
BEGIN
    RAISERROR ('HATA: Girilen Albüm ID sistemde kayıtlı değil. Geçerli bir albüm
seçmelisiniz.', 16, 1);
    ROLLBACK TRANSACTION;
    RETURN;
END

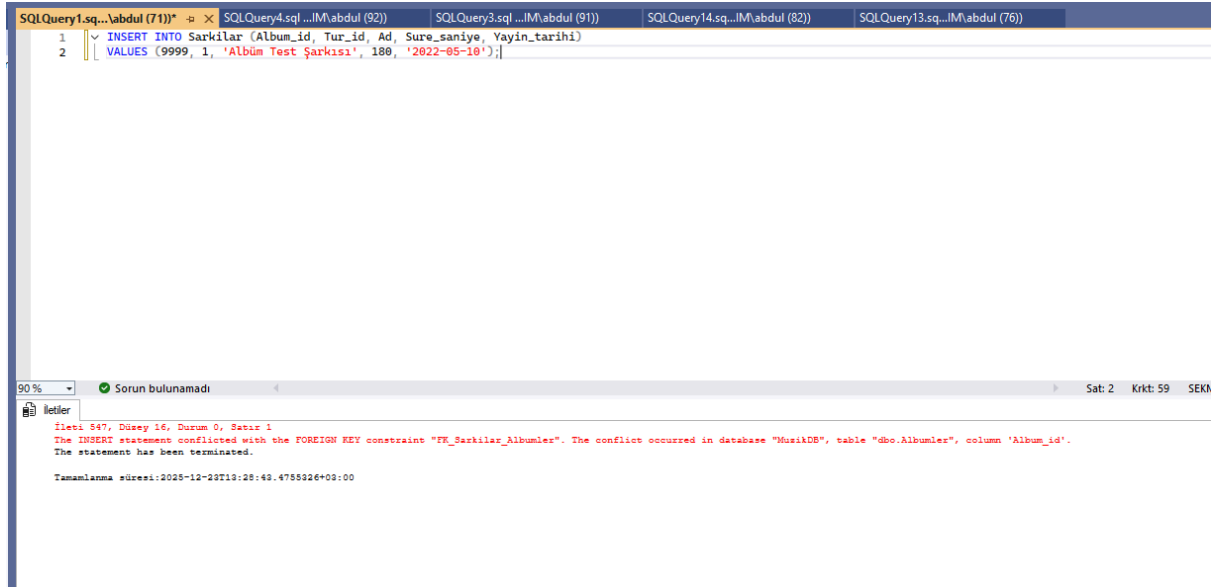
IF EXISTS (
    SELECT i.Sarki_id
    FROM inserted i
    LEFT JOIN Turler t ON i.Tur_id = t.Tur_id
    WHERE t.Tur_id IS NULL -- Eşleşen bir tür bulunamadıysa
)
BEGIN
    RAISERROR ('HATA: Girilen Tür ID sistemde kayıtlı değil. Geçerli bir müzik türü
seçmelisiniz.', 16, 1);
    ROLLBACK TRANSACTION;
    RETURN;
END
END;
GO
```

### 9.2.3 Test Senaryoları ve Sonuçlar

Albüm veya Tür Bilgisi Olmadan Şarkı Ekleme



## Sistemde Olmayan Albüm ID ile Şarkı Ekleme



## Sistemde Olmayan Tür ID ile Şarkı Ekleme



SQLQuery1.sql...Abdul (71)) SQLQuery4.sql...IM\Abdul (92)) SQLQuery3.sql...IM\Abdul (91)) SQLQuery14.sql...IM\Abdul (82)) SQLQuery13.sql...IM\Abdul (76))

```

1  INSERT INTO Sarkilar (Album_id, Tur_id, Ad, Sure_saniye, Yayin_tarihi)
2  VALUES (1, 9999, 'Tür Test Şarkısı', 210, '2021-08-15');

```

90 % Sorun bulunamadı Sat: 2 Krkt: 57 SEKMELER

Betler

Msg 547, Düzey 16, Durum 0, Satır 1:  
The INSERT statement conflicted with the FOREIGN KEY constraint "FK\_Sarkilar\_Turler". The conflict occurred in database "MusikDB", table "dbo.Turler", column "Tur\_id".  
The statement has been terminated.

Tamamlanma süresi: 2025-12-20T19:29:43.1817092+03:00

## Geçerli Albüm ve Tür ile Şarkı Ekleme

SQLQuery1.sql...Abdul (71)) SQLQuery4.sql...IM\Abdul (92)) SQLQuery3.sql...IM\Abdul (91)) SQLQuery14.sql...IM\Abdul (82)) SQLQuery13.sql...IM\Abdul (76))

```

1  INSERT INTO Sarkilar (Album_id, Tur_id, Ad, Sure_saniye, Yayin_tarihi)
2  VALUES (1, 1, 'Başarılı Test Şarkısı', 240, '2020-06-20');

```

90 % Sorun bulunamadı Sat: 2 Krkt: 59 SEKME

Betler

(1 satır ekleildi)

Tamamlanma süresi: 2025-12-20T19:30:47.3810979+03:00

## 10. TRANSACTION YÖNETİMİ

### 10.1 Transaction Senaryosu Adı

#### Çalma Listesi Oluşturma ve Şarkı Ekleme İşlemi

Bu senaryoda bir kullanıcı için yeni bir çalma listesi oluşturulmakta ve oluşturulan bu çalma listesine bir şarkı eklenmektedir.

Bu iki işlem birbirine bağlıdır ve **atomik** olarak gerçekleştirilmelidir.

- Çalma listesi oluşturulup şarkı eklenemezse, çalma listesi de kaydedilmemelidir.
- Tüm işlemler başarılı olursa veritabanına kalıcı olarak kaydedilmelidir.

Bu nedenle işlemler **transaction** içerisinde yönetilmiştir.

---

### 10.2 Transaction Kodu

Aşağıdaki SQL kodunda transaction başlatılmakta, işlemler başarıyla tamamlanırsa **COMMIT**, hata oluşursa **ROLLBACK** yapılmaktadır.

```
BEGIN TRANSACTION;
```

```
BEGIN TRY
```

```
    INSERT INTO CalmaListeleri (Kullanici_id, Liste_adi)  
    VALUES (1, 'Favorilerim');
```

```
    DECLARE @CalmaListesiID INT = SCOPE_IDENTITY();
```

```
    INSERT INTO CalmaListesiSarkilari (Calmalistesi_id,  
Sarki_id, Sira)  
    VALUES (@CalmaListesiID, 1, 1);
```

```
    COMMIT TRANSACTION;
```

```
END TRY
```

```
BEGIN CATCH
    ROLLBACK TRANSACTION;
END CATCH;
```

---

### 10.3 Başarılı Senaryo Testi (COMMIT)

#### İşlemler:

- Kullanici\_id = 1 olan kullanıcı mevcuttur.
- Sarki\_id = 1 olan şarkı mevcuttur.
- Tüm foreign key kısıtları sağlanmaktadır.

#### Sonuç:

- CalmaListeleri tablosuna yeni bir çalma listesi eklenir.
  - Aynı anda CalmaListesiSarkilari tablosuna ilgili şarkı eklenir.
  - Transaction başarıyla tamamlanır ve **COMMIT** edilir.
  - Veriler kalıcı olarak veritabanına kaydedilir.
- 

### 10.4 Hata Senaryosu ve ROLLBACK Testi

#### İşlemler:

- Sarki\_id = 1 olmayan bir değer (örneğin 999) ile şarkı eklenmeye çalışılsın.

```
INSERT INTO CalmaListesiSarkilari (Calmalistesi_id, Sarki_id,
Sira)
VALUES (@CalmaListesiID, 999, 1);
```

### **Oluşan Durum:**

- Foreign key kısıtı ihlal edilir.
- SQL Server hata üretir.
- BEGIN CATCH bloğu çalışır.
- **ROLLBACK TRANSACTION** ile tüm işlemler geri alınır.

### **Sonuç:**

- Çalma listesi **oluşturulmaz**.
- Şarkı ekleme işlemi gerçekleşmez.
- Veritabanı tutarlı (consistent) halde kalır.

## 10.1 Transaction İşlemi Adı

### Çalma Listesi Oluşturma ve Çalma Listesine Şarkı Ekleme İşlemi

Bu senaryoda bir kullanıcı için yeni bir çalma listesi oluşturulmakta ve bu çalma listesine birden fazla şarkı eklenmektedir.

Çalma listesi oluşturulup şarkılar eklenemezse, sistemde **boş veya eksik çalma listesi oluşmaması** gerekir.

Bu nedenle:

- Çalma listesi ekleme
- Çalma listesine şarkı ekleme

işlemleri **tek bir transaction** içerisinde gerçekleştirilmelidir.

---

## 10.2 Transaction Kodu

Aşağıdaki SQL kodunda, çalma listesi oluşturma ve şarkı ekleme işlemleri **BEGIN TRANSACTION** ile başlatılmış, işlemler başarıyla tamamlanırsa **COMMIT**, hata oluşursa **ROLLBACK** yapılmıştır.

```
BEGIN TRANSACTION;
```

```
BEGIN TRY
```

```
    INSERT INTO CalmaListeleri (Kullanici_id, Liste_adi)  
    VALUES (1, 'Ahmet Favori');
```

```
    DECLARE @CalmaListesiID INT = SCOPE_IDENTITY();
```

```
    INSERT INTO CalmaListesiSarkilari (Calmalistesi_id,  
Sarki_id, Sira)  
    VALUES
```

```
        (@CalmaListesiID, 15, 1),  
        (@CalmaListesiID, 19, 2),
```

```
(@CalmaListesiID, 35, 3);
```

```
COMMIT TRANSACTION;
```

```
END TRY
```

```
BEGIN CATCH
```

```
ROLLBACK TRANSACTION;
```

```
END CATCH;
```

---

### 10.3 Başarılı Senaryo Testi (COMMIT)

#### İşlemler:

- Kullanici\_id = 1 olan kullanıcı mevcuttur.
- Eklenmek istenen Sarki\_id değerleri veritabanında bulunmaktadır.
- Foreign key kısıtları ihlal edilmemektedir.

#### Sonuç:

- CalmaListeleri tablosuna yeni bir çalma listesi eklenir.
- Aynı çalma listesine ait şarkılar CalmaListesiSarkilari tablosuna eklenir.
- Tüm işlemler başarıyla tamamlanır ve **COMMIT** edilir.
- Veriler kalıcı olarak veritabanına yazılır.

---

### 10.4 Hata Senaryosu ve ROLLBACK Testi

#### İşlemler:

Çalma listesine eklenmek istenen şarkılardan biri veritabanında yoktur (örneğin Sarki\_id = 999).

```
INSERT INTO CalmaListesiSarkilari (CalmaListesi_id, Sarki_id,  
Sira)  
VALUES (@CalmaListesiID, 999, 4);
```

#### **Oluşan Durum:**

- Foreign key kısıtı ihlal edilir.
- SQL Server hata üretir.
- BEGIN CATCH bloğu çalışır.
- **ROLLBACK TRANSACTION** ile tüm işlemler geri alınır.

#### **Sonuç:**

- Çalma listesi oluşturulmaz.
  - Şarkı ekleme işlemleri iptal edilir.
  - Veritabanı tutarlı (consistent) halde kalır.
-

## 11. TAKIM ÇALIŞMASI VE GÖREV DAĞILIMI

### 11.1 Takım Üyeleri ve Roller

Takım Üyesi	Öğrenci No	Temel Rol
Şeyma Nur ŞAHİN	235260024	Proje Lideri / Veri Tabanı Tasarımcısı
Abdulkerim YILMAZ	235260030	SQL Geliştirici /Dokümantasyon
Ziya EKİZ	235260016	Test / Kalite Kontrol

### 11.2 Gerçekleştirilen İşler ve Sorumluluk Matrisi

No	Yapılan İş / Özellik	Talimat Veren	Gerçekleştiren	Kontrol Eden	Durum
1	Proje gereksinimlerinin belirlenmesi	Şeyma Nur ŞAHİN	Şeyma Nur ŞAHİN	ABDULKERİM YILMAZ	✓
2	E-R diyagramının oluşturulması	Şeyma Nur ŞAHİN ABDULKERİM YILMAZ ZİYA EKİZ	Şeyma Nur ŞAHİN ABDULKERİM YILMAZ ZİYA EKİZ	Şeyma Nur ŞAHİN ABDULKERİM YILMAZ ZİYA EKİZ	✓
3	İlişkisel şemalara dönüştürme	ZİYA EKİZ	ZİYA EKİZ	Şeyma Nur ŞAHİN	✓
4	Normalizasyon (3NF/BCNF)	ŞEYMA NUR ŞAHİN	ŞEYMA NUR ŞAHİN	ABDULKERİM YILMAZ	✓
5	SQL Server'da şema oluşturma	ŞEYMA NUR ŞAHİN	ŞEYMA NUR ŞAHİN	ABDULKERİM YILMAZ,ZİYA EKİZ	✓
6	Birincil/Yabancı anahtar tanımları	ŞEYMA NUR ŞAHİN	ŞEYMA NUR ŞAHİN	ZİYA EKİZ	✓
7	Nitelik kısıtlamalarının eklenmesi	ŞEYMA NUR ŞAHİN	ŞEYMA NUR ŞAHİN	ZİYA EKİZ	✓
8	Örnek verilerin eklenmesi	ABDULKERİM YILMAZ	ABDULKERİM YILMAZ	Şeyma Nur ŞAHİN	✓
9	SQL sorgu script dosyalarının hazırlanması	ABDULKERİM YILMAZ	ABDULKERİM YILMAZ	Şeyma Nur ŞAHİN	✓



10	Saklı yordamın (Stored Procedure) yazılması	ŞEYMA NUR ŞAHİN	ŞEYMA NUR ŞAHİN	ABDULKERİM YILMAZ,ZİYA EKİZ	✓
11	Tetikleyicinin (Trigger) yazılması	ABDULKERİM YILMAZ	ABDULKERİM YILMAZ	ABDULKERİM YILMAZ,ŞEYMA NUR ŞAHİN	✓
12	Transaction yönetimi kodunun yazılması	ZİYA EKİZ	ZİYA EKİZ	ABDULKERİM YILMAZ,ŞEYMA NUR ŞAHİN	✓
13	COMMIT/ROLLBACK test senaryoları	ZİYA EKİZ	ZİYA EKİZ	ABDULKERİM YILMAZ,ŞEYMA NUR ŞAHİN	✓
14	Rapor hazırlama ve dokümantasyon	ŞEYMA NUR ŞAHİN ABDULKERİM YILMAZ ZİYA EKİZ	ŞEYMA NUR ŞAHİN ABDULKERİM YILMAZ ZİYA EKİZ	ŞEYMA NUR ŞAHİN ABDULKERİM YILMAZ ZİYA EKİZ	✓
15	Son kontrol ve teslim	ŞEYMA NUR ŞAHİN ABDULKERİM YILMAZ ZİYA EKİZ	ŞEYMA NUR ŞAHİN ABDULKERİM YILMAZ ZİYA EKİZ	ŞEYMA NUR ŞAHİN ABDULKERİM YILMAZ ZİYA EKİZ	✓

**Not:** Talimat Veren: İlgili işin yapılması gerektiğine karar veren ve yönlendiren kişi.  
Gerçekleştiren: İş fiilen yapan kişi. Kontrol Eden: İşin doğruluğunu ve kalitesini kontrol eden kişi.

## 12. SONUÇ VE DEĞERLENDİRME

Projemizde sql komutlarını nasıl kullanıp onları daha verimli ve optimize çalıştırmayı öğrendik bunları yaparken aynı zamanda nasıl veri eklenir (kendimiz veri seti oluşturup bu veriler doğrultusunda ekleme yaptık), veri silinir, güncellenir bunları kontrol edip kullanmayı öğrendik. Normalizasyon kavramlarının nasıl kullanacağı ve kısıtların oluşturduğu etkileri inceleyip ayrımlarını öğrendik. Daha sonrasında Stored procedure kullanımını üzerinde durduk ve Triggerları nasıl ekleyeceğimizi öğrendik. Sonrasında Transaction hata yönetiminin nasıl yapılıp çözüleceğini en hızlı ve en verimli şekilde çözmeyi öğrendik. Commit ile yaptığımız işlemleri test edip en doğru ve uygun sonuç almayı hedefledik.