

MÜZİK PLATFORMU VERİ TABANI

İçindekiler

DİYAGRAM.....	8
NORMALİZASYON	9
TABLOLAMA	10
SORGULAR.....	11
VERİ TABANINA BAĞLANTI KODU	25
FORM1.CS.....	25
FORM2.CS.....	26
FORM4.CS.....	27
TÜMSANATÇILAR.CS	28
TÜMSARKILAR.CS.....	28
CALMALİSTELERİ.CS	29
CALMALİSTESİOLUSTUR.CS	29
RASTGELE ŞARKI EKLEME.....	30
YETMİŞLER.CS	31
SEKSENLER.CS	31
DOKSANLAR.CS.....	32
İKİBİNLER.CS	32

Önsöz

Bu rapor, Müzik Platformu projesinin tasarımı, geliştirilmesi ve uygulanması sürecini detaylı bir şekilde ele almaktadır. Bu proje, MSSQL veri tabanı kullanılarak oluşturulan bir müzik platformunun yapısını ve işlevselliğini anlatmaktadır.

Bu projenin amacı, modern bir müzik platformu oluştururken veri tabanı yönetiminin önemini vurgulamaktır. MSSQL kullanarak veri tabanı oluşturma, veri modelleme, sorgu oluşturma ve veri güvenliği gibi konular bu raporda detaylı bir şekilde incelenmiştir.

Raporun ilk bölümünde, proje gereksinimleri ve hedefleri açıklanmıştır. Ardından, veri tabanı tasarımı adımları ayrıntılı bir şekilde anlatılmıştır. Veri tabanı oluşturma sürecinde kullanılan tablolar, ilişkiler ve sorgular ayrıntılı olarak açıklanmıştır.

Kullanım Amacı

Bu rapor, MSSQL veri tabanı kullanılarak geliştirilen müzik platformu projesinin detaylı bir incelemesini sunmayı amaçlamaktadır. Projede kullanılan veri tabanı tasarımı, işlevselliği ve güvenlik önlemleri üzerinde durularak, benzer veri tabanı projeleri geliştirme sürecinde faydalı bir kaynak oluşturmayı hedeflemektedir. Ayrıca, rapor, veri tabanı yönetimi konusunda geniş bir anlayış sağlamak ve bu alanda çalışanlara, öğrencilere ve ilgili kişilere rehberlik etmek için hazırlanmıştır.

Görev Dağılımı

Nisa Usta – Otomasyon Geliştirme / Normalizasyon / Veri Girişi

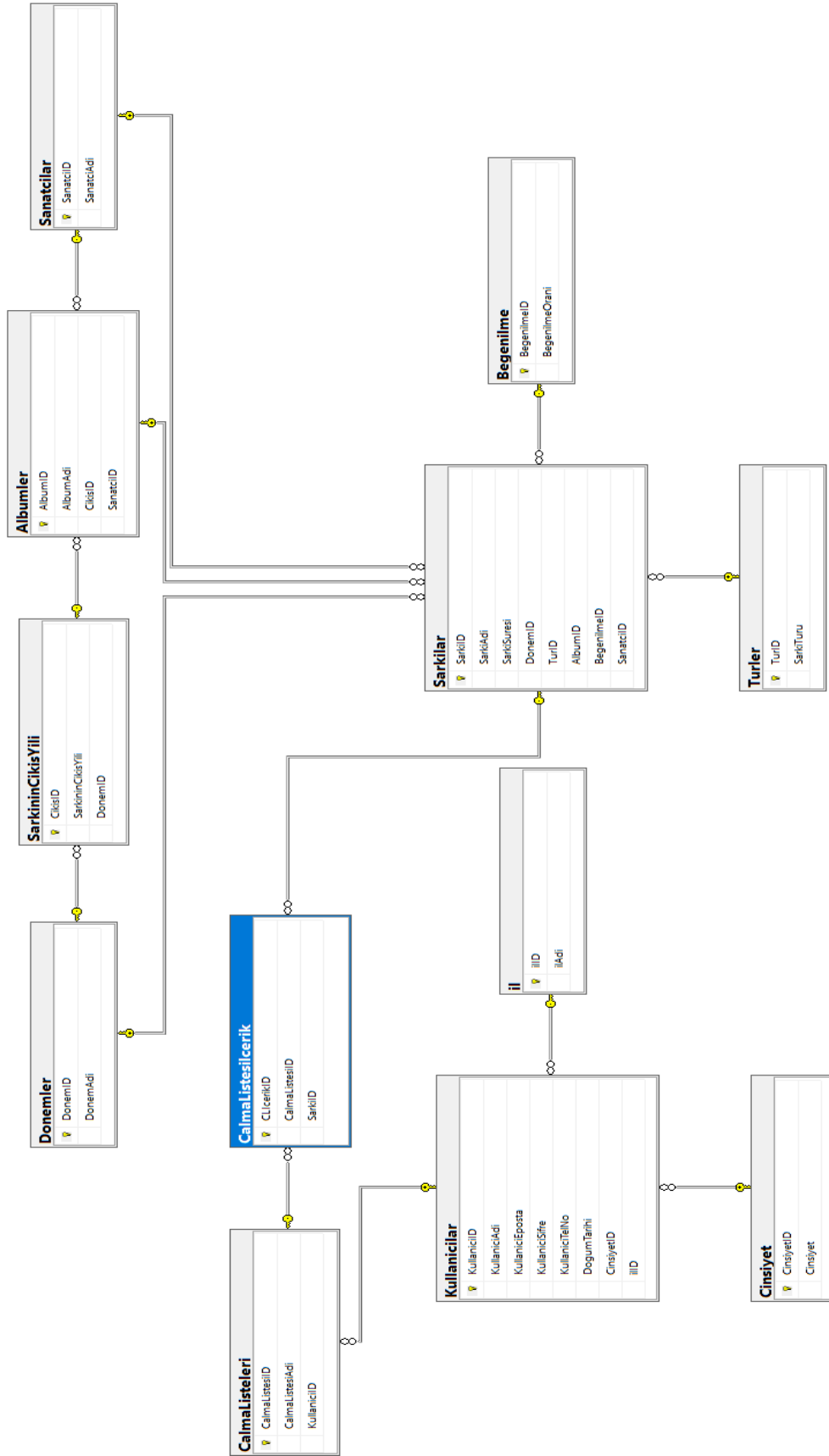
Muhammed Rıdvan Beyiş - Otomasyon Geliştirme / Normalizasyon / Veri Girişi

Şimal Bülbül – Veri Girişi / Normalizasyon / Sorgular / Tablo Oluşturma

Ceydanur Özgür- Veri Toplama / Sorgular / Veri Girişi

Eray Güngör - Veri Toplama / Sorgular / Veri Girişi

DIYAGRAM



NORMALİZASYON

SANATÇILAR		ALBÜMLER			
SanatciID	SanatciAdi	AlbumID	AlbumAdi	CikisID	SanatciID
1	AA	1	A	1	1
2	AA	2	A	2	2
3	AA	3	A	2	3
4	AA	4	A	3	4
5	AA	5	A	5	5

ŞARKILAR						
SarkiID	SarkiAdi	SarkiSuresi	TurID	AlbumID	DonemID	BegenilmelID
1	A	454	1	1	1	1
2	A	454	5	2	2	2
3	A	45	5	3	3	3
4	A	555	5	4	4	4
5	A	555	2	5	2	5

ÇALMA LİSTESİ			BEĞENİLME		
CalmaListesiID	ÇALMA LİSTESİADİ	KullaniciID	BegenilmelID	BegenilmeOrani	
1	A	1	1	1	
2	A	2	2	1,1	
3	A	3	3	1,2	
4	A	4			
5	A	5			

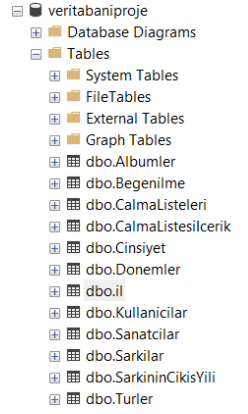
ŞARKININ ÇIKIŞ YILI		
CikisID	SarkininCikisYili	DonemID
1	1990	3
2	2000	4
3	1985	2
4	1975	1
5	1965	2

ÇALMA LİSTESİ İÇERİK			CİNSİYET	
İçerikID	SarkiID	CalmaListesiID	CinsiyetID	KullaniciCinsiyet
1	1	1	1	kadın
2	2	2	2	erkek
3	3	3	3	diğer
4	4	4		
5	5	5		

TÜRLER		İLLER		DÖNEMLER	
TurID	SarkiTuru	ilID	ilAd	DonemID	Donemler
1	POP	1	ist	1	70'ler
2	ROCK	2	artvin	2	80'ler
3	HALK MÜZ	3	kars	3	90'lar
4	TÜRK SN	4	rize	4	00'lar
5	METALİCA	5	ordu		

TABLOLAMA

Veritabaniproje adında veri tabanı oluşturduktan sonra tablo girişleri yapıldı.



Sonrasında tablolama oluşturduktan sonra sorgular ile veri girişi yapıldı. Bunlarda sorgular:

SORGULAR

-- SELECT sorguları : Tablo Listeler

```
SELECT * FROM Sanatcilar;
```

	SanatciID	SanatciAdi
▶	1	Ahmet
	2	Ajda Pekkan
	3	Aşkın Nur Y...
	4	Ayşe Hatun
	5	Barış Akarsu

-- INSERT sorguları Sanatcilar tablosuna veri girişi yapmayı sağlar.

```
INSERT INTO Sanatcilar (SanatciID, SanatciAdi) VALUES (51, 'Seyfettin Sucu');
```

51	51	Seyfettin Sucu
----	----	----------------

-- UPDATE sorguları : Güncellemeyi sağlar.

```
UPDATE Sanatcilar SET SanatciAdi = 'Emel Taşçıoğlu' WHERE SanatciID = 51;
```

51	Emel Taşçıoğlu
----	----------------

-- WHERE koşulları : Koşul belirtir.

```
SELECT * FROM Sanatcilar WHERE SanatciAdi = 'Ajda Pekkan';
```

2	Ajda Pekkan
---	-------------

-- JOIN işlemleri: İki tablonun belirli bir koşula göre eşleşmesini sağlar.

```
SELECT * FROM Sarkilar INNER JOIN Sanatcilar ON Sarkilar.SanatciID = Sanatcilar.SanatciID;
```

1	Sonsuz Çile	03:42:00	3	4	1	15	1	1	Ahmet
---	-------------	----------	---	---	---	----	---	---	-------

-- GROUP BY ve HAVING kullanımı : Sütuna göre veri gruplama ve grupların toplu işlemlerini filtreleme

```
SELECT SanatciID, COUNT(*) AS 'Toplam Şarkı' FROM Sarkilar GROUP BY SanatciID HAVING COUNT(*) > 1;
```

SanatciID	Toplam Şarkı
2	3

-- ORDER BY kullanımı : Bir tablodan tüm sütunları alır ancak order by ile belirtilen sütuna göre sıralar

```
SELECT * FROM Sarkilar ORDER BY SarkiAdi;
```

SarkiID	SarkiAdi	SarkiSuresi	DonemID	TurlID	AlbumID	BegenilmeID	SanatciID
17	Adını Suya Yazdım	03:45:00	3	4	15	26	10

-- COUNT: Satır sayısını döndürür. SUM:Belirli sütundaki değerleri toplar.
AVG: Ortalamayı bulur.

```
SELECT COUNT(*) AS 'Şarkılar' FROM Sarkilar; SELECT AVG(BegenilmeOrani) AS  
'Beğenilme Oranı' FROM BEGENİLME; SELECT SUM(BegenilmeOrani) AS 'Beğenilme  
Puan Toplamı' FROM BEGENİLME;
```

(50 rows affected)

Completion time: 2023-12-29T20:52:18.9610303+03:00

-- Bir kayan noktalı sayı tipindeki değişken tanımla: @maxSarkiSuresi ///
Değişken bildirimi yapmamıza yarar.

```
DECLARE @maxSarkiSuresi FLOAT;
```

(1 row affected)

Completion time: 2023-12-29T21:02:17.1690531+03:00

UPDATE Albumler

```
SET AlbumAdi = (SELECT TOP 1 AlbumAdi FROM Albumler ORDER BY AlbumID DESC), --  
Albumler tablosundaki AlbumAdi sütununu, aynı tablodaki en yüksek AlbumID  
değerine sahip AlbumAdi ile güncelleme
```

```
    CikisID = (SELECT TOP 1 CikisID FROM Albumler ORDER BY AlbumID DESC), --  
Albumler tablosundaki CikisID sütununu, aynı tablodaki en yüksek AlbumID  
değerine sahip CikisID ile güncelleme
```

```
    SanatciID = (SELECT TOP 1 SanatciID FROM Albumler ORDER BY AlbumID DESC) -  
- Albumler tablosundaki SanatciID sütununu, aynı tablodaki en yüksek AlbumID  
değerine sahip SanatciID ile güncelleme
```

```
WHERE AlbumID = (SELECT MAX(AlbumID) FROM Albumler); -- Güncellemeyi hangi  
kriterlere göre yapmak istediğiniz
```


	AdID
1	Nisa 1
2	Rıdvan 2
3	Şimal 3
4	Eray 4
5	Ceyda 5

```
SELECT KullaniciAdi + ' ' + CONVERT(nvarchar, KullaniciID) AS 'AdID'
FROM Kullanicilar; // "Kullanicilar" tablosundaki her bir kullanıcı için,
"KullaniciAdi" ve "KullaniciID" alanlarını birleştirerek yeni bir sütun
oluşturur. Oluşturulan bu yeni sütunun adı 'AdID' olarak belirlenmiştir.
```

```
SELECT AlbumID, SUM(DATEDIFF(minute, '00:00:00', SarkiSuresi)) AS
'ToplamDakika'
FROM Sarkilar
GROUP BY AlbumID
HAVING SUM(DATEDIFF(minute, '00:00:00', SarkiSuresi)) > 150
ORDER BY ToplamDakika ASC;
```

	(No column name)
1	300

	(No column name)
1	302

```
DECLARE @degisken1 INT; -- Tamsayı bir değişken tanımla.
SET @degisken1 = 300;-- Değişken değerine 300 atama yap.

IF @degisken1 > 300 -- Eğer değişken değeri 300'den büyükse
BEGIN
    SELECT MAX(CikisID) FROM Albumler; -- "Albumler" tablosundaki en büyük
    CikisID değerini seç.
    SELECT MIN(CikisID) FROM Albumler; -- "Albumler" tablosundaki en küçük
    CikisID değerini seç.

    SET @degisken1 = @degisken1 + 1;-- Değişken değerini bir artır.
END
ELSE
BEGIN
    SELECT @degisken1;-- Değişkenin mevcut değerini seç.
    SET @degisken1 = @degisken1 + 2;-- Değişken değerini iki artır.
    SELECT @degisken1;-- Değişkenin güncellenmiş değerini seç.
END;
```

Commands completed successfully.

Completion time: 2023-12-29T21:03:40.5233308+03:00

```
DECLARE @degisken1 INT; -- Tamsayı bir değişken tanımla.  
SET @degisken1 = 300;-- Değişken değerine 300 atama yap.
```

```
-- "Sanatcilar", "Sarkilar", "Turler", ve "Begenilme" tablolarını  
birleştirerek her sanatçının her türdeki en düşük beğenilme oranını bul.  
-- Gruplama yapılarak, her sanatçı ve tür kombinasyonu için en düşük beğenilme  
oranını içeren sonuçları elde et.  
-- ROLLUP kullanarak toplam (overall) en düşük beğenilme oranlarını da göster.  
-- Sonuçları tür ID'ye göre artan sırayla sırala.  
SELECT Sanatcilar.SanatciAdi, Turler.TurID, MIN(Begenilme.BegenilmeOrani) AS  
MinBegenilmeOrani  
FROM Sanatcilar  
JOIN Sarkilar ON Sanatcilar.SanatciID = Sarkilar.SanatciID  
JOIN Turler ON Sarkilar.TurID = Turler.TurID  
JOIN Begenilme ON Sarkilar.BegenilmeID = Begenilme.BegenilmeID  
GROUP BY Sanatcilar.SanatciAdi, Turler.TurID WITH ROLLUP  
ORDER BY Turler.TurID ASC;
```

	(No column name)
1	temhA
2	nakkeP adjA
3	igneY ruN nıkşA
4	nutaH eşyA
5	usrakA şıraB
6	oçnaM şıraB
7	üğneB
8	negreB
9	acaraK meC
10	rezÖ meC
11	nusaT meC

```
-- "Sanatcilar" tablosundaki "SanatciAdi" sütunundaki değerleri ters çevir.  
select reverse(SanatciAdi) from Sanatcilar
```

	SanatciAdi
1	Ahmet
2	Ajda Pekkan
3	Aşkın Nur Yengi
4	Ayşe Hatun
5	Barış Akarsu
6	Barış Manço
7	Bengü
8	Bergen
9	Cem Karaca
10	Cem Özer
11	Cem Torun

```
-- "Sanatcilar" tablosundaki benzersiz (tekrarsız) "SanatciAdi" değerlerini seç.
select distinct(SanatciAdi) from Sanatcilar
```

	tum_yillar
1	40

```
-- "SarkininCikisYili" tablosundaki toplam satır sayısını bul.
-- Bu sayıyı "tum_yillar" ismiyle alias (takma ad) ile görüntüle.
select count(SarkininCikisYili) as 'tum_yillar' from SarkininCikisYili
```

	ort_yil
1	1989

```
-- "SarkininCikisYili" tablosundaki "SarkinCikisYili" sütunundaki değerlerin ortalamasını hesapla.
-- Ortalamayı "ort_yil" ismiyle alias (takma ad) ile görüntüle.
select avg(SarkininCikisYili) as 'ort_yil' from SarkininCikisYili
```

	top_yil
1	79580

```
-- "SarkininCikisYili" tablosundaki "SarkinCikisYili" sütunundaki tüm değerleri topla.
-- Toplamı "top_yil" ismiyle alias (takma ad) ile görüntüle.
select sum(SarkininCikisYili) as 'top_yil' from SarkininCikisYili
```

	min_cikis_yili
1	1970

```
-- "SarkinCikisYili" tablosundaki "SarkinCikisYili" sütunundaki en düşük
değeri (minimum) bul.
-- Bu değeri "min_cikis_yili" ismiyle alias (takma ad) ile görüntüle.
select min(SarkinCikisYili) as 'min_cikis_yili' from SarkinCikisYili
```

	max_cikis_yili
1	2009

```
-- "SarkinCikisYili" tablosundaki "SarkinCikisYili" sütunundaki en yüksek
değeri (maksimum) bul.
-- Bu değeri "max_cikis_yili" ismiyle alias (takma ad) ile görüntüle.
select max(SarkinCikisYili) as 'max_cikis_yili' from
SarkinCikisYili
```

	CikisID	sarkinincikisyili	DonemID
1	20	1989	2
2	19	1988	2
3	18	1987	2
4	17	1986	2
5	16	1985	2
6	15	1984	2
7	14	1983	2
8	13	1982	2
9	12	1981	2
10	11	1980	2

```
-- "SarkinCikisYili" tablosundan, "DonemID" değeri 2 olan kayıtların
"CikisID", "sarkinincikisyili", ve "DonemID" sütunlarını seç.
-- Sonuçları "Sarkinincikisyili" sütununa göre büyükten küçüğe doğru (azalan
sıra) sırala.
select CikisID, sarkinincikisyili, DonemID from SarkinCikisYili where DonemID
= 2 order by SarkinCikisYili desc
```

	CikisID	sarkinincikisyili	DonemID
1	1	1970	1
2	2	1971	1
3	3	1972	1
4	4	1973	1
5	5	1974	1
6	6	1975	1
7	7	1976	1
8	8	1977	1
9	9	1978	1
10	10	1979	1

```
-- "SarkinCikisYili" tablosundan, "DonemID" değeri 1 olan kayıtların
"CikisID", "sarkinincikisyili", ve "DonemID" sütunlarını seç.
-- Sonuçları "Sarkinincikisyili" sütununa göre küçükten büyüğe doğru (artan
sıra) sırala.
select CikisID, sarkinincikisyili,DonemID from SarkinCikisYili where DonemID
= 1 order by SarkinCikisYili asc
```

	Aradaki Fark
1	0

```
--1980-1990 arası en çok şarkı çıkaran sanatçı ile 1990-2000 arası en çok
şarkı çıkartan sanatçının arasındaki şarkı sayısı farkı
SELECT COUNT(*) AS 'Aradaki Fark' FROM Sanatcilar
INNER JOIN ALBUMLER ON Sanatcilar.SanatciID = ALBUMLER.SanatciID
INNER JOIN SarkinCikisYili ON ALBUMLER.CikisID = SarkinCikisYili.CikisID
INNER JOIN Sarkilar ON ALBUMLER.AlbumID = Sarkilar.AlbumID WHERE
SarkinCikisYili.DonemID = 1 OR SarkinCikisYili.DonemID = 2
```

	SanatciID	SanatciAdi
1	5	Baş Akarsu
2	6	Baş Manço
3	7	Bengü
4	8	Bergen
5	14	Edip Akbayram
6	22	Gülben Ergen
7	46	Suavi Karabrahimgil
8	47	Şebnem Aksu

```
--CHARINDEX Kullanımı
SELECT * FROM Sanatcilar WHERE CHARINDEX('B', SanatciAdi) > 0;
```

	Karesi Alınmış Beğenilme Oranı
1	1
2	1,21
3	1,44
4	1,69
5	1,96
6	2,25
7	2,56
8	2,89
9	3,24
10	3,61
11	4

-- POWER: Bir sayının belirli bir kuvvetini döndürür.

```
SELECT POWER(BeğenilmeOranı, 2) AS 'Karesi Alınmış Beğenilme Oranı' FROM BEĞENİLME;
```

	Karekök Beğenilme Oranı
1	1
2	1,04880884817015
3	1,09544511501033
4	1,14017542509914
5	1,18321595661992
6	1,22474487139159
7	1,26491106406735
8	1,30384048104053
9	1,34164078649987
10	1,37840487520902
11	1,4142135623731

-- SQRT: Bir sayının karekökünü döndürür.

```
SELECT SQRT(BeğenilmeOranı) AS 'Karekök Beğenilme Oranı' FROM BEĞENİLME
WHERE BeğenilmeOranı >= 0;
```

	Yukarı Yuvarlanmış Begenilme Oranı	Aşağı Yuvarlanmış Begenilme Oranı
1	1	1
2	2	1
3	2	1
4	2	1
5	2	1
6	2	1
7	2	1
8	2	1
9	2	1
10	2	1
11	2	2

```
-- CEILING ve FLOOR: Bir sayıyı yukarı veya aşağıya yuvarlar.
SELECT CEILING(BegenilmeOrani) AS 'Yukarı Yuvarlanmış Begenilme Oranı',
FLOOR(BegenilmeOrani) AS 'Aşağı Yuvarlanmış Begenilme Oranı' FROM BEGENİLME;
```

	Yuvarlanmış Begenilme Oranı
1	1
2	1
3	1
4	1
5	1
6	2
7	2
8	2
9	2
10	2
11	2

```
-- ROUND: Bir sayıyı en yakın tam sayıya yuvarlar.
SELECT ROUND(BegenilmeOrani,0) AS 'Yuvarlanmış Begenilme Oranı' FROM
BEGENİLME;
```

	Mutlak Beğenilme Oranı
1	1
2	1,1
3	1,2
4	1,3
5	1,4
6	1,5
7	1,6
8	1,7
9	1,8
10	1,9
11	2

```
-- ABS: Bir sayının mutlak değerini döndürür.
SELECT ABS(BeğenilmeOranı) AS 'Mutlak Beğenilme Oranı' FROM BEĞENİLME;
```

	Sanatçı Adı (Trim Uygulanmış)
1	Ahmet
2	Ajda Pekkan
3	Aşkın Nur Yengi
4	Ayşe Hatun
5	Barış Akarsu
6	Barış Manço
7	Bengü
8	Bergen
9	Cem Karaca
10	Cem Özer
11	Cem Torun

```
-- TRIM: Bir stringin başındaki ve sonundaki boşlukları kaldırır.
SELECT TRIM(SanatçıAdı) AS 'Sanatçı Adı (Trim Uygulanmış)' FROM
Sanatçılar;
```

	Sanatçı Adı (a yerine e)
1	ehmet
2	ejde Pekken
3	eşkın Nur Yengi
4	eyşe Hetun
5	Barış ekersu
6	Barış Menço
7	Bengü
8	Bergen
9	Cem Kerece
10	Cem Özer
11	Cem Torun

```
-- REPLACE: Bir stringdeki belirli bir metni başka bir metinle değiştirir.
SELECT REPLACE(SanatçıAdı, 'a', 'e') AS 'Sanatçı Adı (a yerine e)' FROM
Sanatçılar;
```


	Sanatci Adi (İlk 5 Karakter)
1	Ahmet
2	Ajda
3	Aşkın
4	Ayşe
5	Barış
6	Barış
7	Bengü
8	Berge
9	Cem K
10	Cem Ö
11	Cem T

```
-- SUBSTRING: Bir stringin belirli bir bölümünü döndürür.
SELECT SUBSTRING(SanatciAdi, 1, 5) AS 'Sanatci Adi (İlk 5 Karakter)'
FROM Sanatcilar;
```

	Sanatci Adi (Büyük Harf)	Sanatci Adi (Küçük Harf)
1	AHMET	ahmet
2	AJDA PEKKAN	ajda pekkkan
3	AŞKIN NUR YENGİ	aşkın nur yengi
4	AYŞE HATUN	ayşe hatun
5	BARIŞ AKARSU	barış akarsu
6	BARIŞ MANÇO	barış manço
7	BENGÜ	bengü
8	BERGEN	bergen
9	CEM KARACA	cem karaca
10	CEM ÖZER	cem özer
11	CEM TOĞUL	cem toğul

```
-- UPPER ve LOWER: Bir stringi büyük veya küçük harfe çevirir.
SELECT UPPER(SanatciAdi) AS 'Sanatci Adi (Büyük Harf)', LOWER(SanatciAdi) AS
'Sanatci Adi (Küçük Harf)' FROM Sanatcilar;
```

	Sanatci Adi Uzunlugu
1	5
2	11
3	15
4	10
5	12
6	11
7	5
8	6
9	10
10	8
11	0

-- LEN: Bir stringin uzunluğunu döndürür.

```
SELECT LEN(SanatciAdi) AS 'Sanatci Adi Uzunlugu' FROM Sanatcilar;
```

	Sanatci ve Album
1	Ahmet - Ah Canım Vah Canım
2	Ajda Pekkan - Süperstar4
3	Ajda Pekkan - 5.Yıl Önce 10 Yıl Sonra
4	Ajda Pekkan - Süperstar 83
5	Aşkın Nur Yengi - Haydi Söyle
6	Ayşe Hatun - Single
7	Barış Akarsu - Ayrılık Zamansız Gelir
8	Barış Manço - Sahibinden ihtiyaçtan
9	Barış Manço - 21.Yüzyılın Adamı
10	Barış Manço - Değmesin Yağlı Boya
11	Barış Manço - Single

-- CONCAT: İki veya daha fazla stringi birleştirir.

```
SELECT CONCAT(SanatciAdi, ' - ', AlbumAdi) AS 'Sanatci ve Album' FROM
Sanatcilar INNER JOIN ALBUMLER ON Sanatcilar.SanatciID = ALBUMLER.SanatciID;
```


	Şarkılar
1	0

	Beğenilme Oranı
1	2.99512195121951

	Beğenilme Puan Toplamı
1	122.8

```
-- COUNT, SUM, AVG gibi agregat fonksiyonlar
SELECT COUNT(*) AS 'Şarkılar' FROM Sarkilar; SELECT AVG(BegenilmeOrani) AS
'Beğenilme Oranı' FROM BEGENİLME; SELECT SUM(BegenilmeOrani) AS 'Beğenilme
Puan Toplamı' FROM BEGENİLME;
```

	SanatciID	SanatciAdi
1	24	Hadise

	SanatciID	SanatciAdi
1	1	Ahmet
2	2	Ajda Pekkan
3	3	Aşkın Nur Yengi

```
SELECT * FROM Sanatcilar WHERE SanatciAdi LIKE '%Adi%'; SELECT * FROM
Sanatcilar WHERE SanatciID IN (1, 2, 3);
```

VERİ TABANINA BAĞLANTI KODU

Veri tabanına bağlıyoruz bu kodda

```
static string conString = "Data Source=NISA;Initial  
Catalog=veritabaniproje;Integrated Security=True";  
SqlConnection connect = new SqlConnection(conString);
```

FORM1.CS

iki farklı butonun bulunduğu bir formda, her bir butonun tıklanması durumunda bir sonraki forma geçişi sağlayan olayları içerir. button1_Click metodu, Form2'yi çağırarak gösterir ve mevcut formu gizler (this.Hide()). Benzer şekilde, button2_Click metodu, Form3'ü çağırarak gösterir ve mevcut formu gizler. Bu şekilde, kullanıcı bir butona tıkladığında ilgili form görüntülenir ve mevcut form gizlenir.

```
private void button1_Click(object sender, EventArgs e)  
{  
    Form2 f = new Form2();  
    f.Show();  
    this.Hide();  
}  
  
private void button2_Click(object sender, EventArgs e)  
{  
    Form3 f = new Form3();  
    f.Show();  
    this.Hide();  
}
```

FORM2.CS

Kullanıcının bir Windows Forms uygulamasında giriş yaptığı bilgileri SQL Server veritabanına eklemek için kullanılır. Kullanıcının adını, e-posta adresini, şifresini ve telefon numarasını içeren bilgiler, "Kullanıcılar" adlı tabloya eklenir. Ekleme işlemi başarılı bir şekilde gerçekleştirildiğinde kullanıcıya bir bilgi mesajı gösterilir. Ayrıca, kod hataya karşı try-catch bloğu içinde yönetilir ve hata durumunda kullanıcıya bir hata mesajı gösterilir. Son olarak, kayıt ekleme işlemi tamamlandıktan sonra Form3'e geçiş yapılır ve mevcut form gizlenir.

```
private void btn_kayitol_Click(object sender, EventArgs e)
{
    try
    {
        if (connect.State == ConnectionState.Closed)
            connect.Open();
        string kayit = "insert into Kullanıcılar
(KullaniciAdi,KullaniciEposta,KullaniciSifre,KullaniciTelNo)
values(@KullaniciAdi,@KullaniciEposta,@KullaniciSifre,@KullaniciTelNo)";
        SqlCommand komut = new SqlCommand(kayit, connect);

        // komut.flarameters.AddWithValue("@KullaniciID",
txt_id.Text);
        komut.flarameters.AddWithValue("@KullaniciAdi",
txt_ad.Text);
        komut.flarameters.AddWithValue("@KullaniciEposta",
txt_eposta.Text);
        komut.flarameters.AddWithValue("@KullaniciSifre",
txt_parola.Text);
        komut.flarameters.AddWithValue("@KullaniciTelNo",
txt_telno.Text);
        //komut.flarameters.AddWithValue("@DogumTarihi",
combobox_dt.Text);
        //komut.flarameters.AddWithValue("@CinsiyetID",
combobox_cinsiyet.Text);
        // komut.flarameters.AddWithValue("@ilID",
combobox_sehir.Text);
        komut.ExecuteNonQuery();

        MessageBox.Show("kayıt eklendi");
        connect.Close();

    }
    catch (Exception hata)
    {
        MessageBox.Show("HATA" + hata.Message);
    }
    Form3 f = new Form3();
    f.Show();
    this.Hide();
}
```

FORM4.CS

kullanıcının giriş yaptığı kullanıcı adı ve şifreyi kontrol ederek, bu bilgilerin doğru olup olmadığını SQL Server veritabanında bulunan "Kullanıcılar" tablosunda sorgular. Eğer giriş bilgileri doğruysa, kullanıcının kimliğini temsil eden "KullaniciID" değerini alır, bu bilgiyi bir özellik olarak saklar, bir etikete "Başarılı" mesajını yazar, ve ardından Form4'ü görüntüler. Eğer giriş bilgileri doğru değilse, bir etikete "Başarısız" mesajını yazar. Hata durumları da try-catch bloğu içinde yönetilir ve bağlantı sonlandırılır.

```
try
{
    con.Open();
    com.Connection = con;
    com.CommandText = "SELECT KullaniciID FROM Kullanıcılar
WHERE KullaniciAdi=@kullaniciAdi AND KullaniciSifre=@sifre";
    com.Parameters.AddWithValue("@kullaniciAdi",
kullaniciAdi);
    com.Parameters.AddWithValue("@sifre", sifre);

    object result = com.ExecuteScalar();

    if (result != null)
    {
        // Giriş başarılı, KullaniciID'yi al ve özelliğe ata
        GirişYapanKullaniciID = Convert.ToInt32(result);

        label1.Text = "Başarılı";
        Form4 form = new Form4(); // Form3'ten bir örnek
oluştur
        Form3.GirişYapanKullaniciID =
Convert.ToInt32(result);
        form.Show();
        this.Hide();
    }
    else
    {
        label1.Text = "Başarısız";
    }
}
catch (Exception ex)
{
    MessageBox.Show("Hata: " + ex.Message);
}
finally
{
    con.Close();
}
}
```

TÜMSANATÇILAR.CS

SQL Server veritabanındaki "Sanatcilar" tablosundan tüm sanatçı adlarını çeker ve bu bilgileri bir DataGridView kontrolüne yükleyerek kullanıcıya gösterir.

```
public void kayitlari_getir()
{
    baglan.Open();
    string getir = "SELECT SanatciAdi FROM Sanatcilar";
    SqlCommand komut = new SqlCommand(getir, baglan);
    SqlDataAdapter ad = new SqlDataAdapter(komut); //listeleme
işlemi yapar datagridden

    DataTable dt = new DataTable();
    ad.Fill(dt); //datatable doldur
    dataGridView1.DataSource = dt;

    baglan.Close();
}
```

TÜMSARKILAR.CS

SQL Server veritabanındaki "Sarkilar" tablosu ile ilişkili olan tur, album ve sanatçı bilgilerini içeren tüm şarkı kayıtlarını çeker ve bu bilgileri bir DataGridView kontrolüne yükleyerek kullanıcıya gösterir.

```
public void kayitlari_getir()
{
    baglan.Open();
    string getir = "SELECT Sarkilar.SarkiAdi,
Sarkilar.SarkiSuresi, turler.SarkiTuru, albumler.AlbumAdi,
sanatcilar.SanatciAdi " +
                    "FROM Sarkilar " +
                    "INNER JOIN turler ON Sarkilar.TurID =
turler.TurID " +
                    "INNER JOIN albumler ON Sarkilar.AlbumID =
albumler.AlbumID " +
                    "INNER JOIN sanatcilar ON Sarkilar.SanatciID =
sanatcilar.SanatciID;";
    SqlCommand komut = new SqlCommand(getir, baglan);
    SqlDataAdapter ad = new SqlDataAdapter(komut); //listeleme
işlemi yapar datagridden

    DataTable dt = new DataTable();
    ad.Fill(dt); //datatable doldur
    dataGridView1.DataSource = dt;

    baglan.Close();
}
```


CALMALİSTELERİ.CS

SQL Server veritabanındaki "CalmaListeleri" tablosu ile ilişkilendirilmiş kullanıcı adları ve çalma listesi adlarını içeren bilgileri çeker ve bu bilgileri bir DataGridView kontrolüne yükleyerek kullanıcıya gösterir.

```
public void kayitlari_getir()
{
    baglan.Open();
    string getir = "SELECT CalmaListeleri.CalmalistesiAdi,
KullaniciAdi " +
        "FROM CalmaListeleri " +
        "INNER JOIN kullanicilar ON
CalmaListeleri.KullaniciID = Kullanicilar.KullaniciID;";
    SqlCommand komut = new SqlCommand(getir, baglan);
    SqlDataAdapter ad = new SqlDataAdapter(komut); //listeleme
işlemi yapar datagridden
    DataTable dt = new DataTable();
    ad.Fill(dt); //datatable doldur
    dataGridView1.DataSource = dt;
    baglan.Close();
}
```

CALMALİSTESİOLUSTUR.CS

Windows Forms uygulamasında bir checkedListBox'ten seçilen şarkıları bir List koleksiyonuna ekleyen btn_ekle_Click ve bu seçilen şarkıları bir DataGridView kontrolünde gösteren DataGridViewGuncelle metodlarını içerir. secilenSarkilar adlı bir List koleksiyonu, checkedListBox'ten seçilen şarkı adlarını saklamak için kullanılır. Her bir seçilen şarkı adı, bu List koleksiyonuna eklenir ve ardından DataGridViewGuncelle metodu çağrılarak DataGridView kontrolü güncellenir ve kullanıcıya seçilen şarkıları gösterir.

```
private void btn_ekle_Click(object sender, EventArgs e)
{
    secilenSarkilar.Clear();

    foreach (var checkedItem in checkedListBox1.CheckedItems)
    {
        secilenSarkilar.Add(checkedItem.ToString());
    }
    DataGridViewGuncelle();
}
private void DataGridViewGuncelle()
{
    DataTable dt = new DataTable();
    dt.Columns.Add("SarkiAdi");

    foreach (var sarkiAdi in secilenSarkilar)
    {
        DataRow row = dt.NewRow();
        row["SarkiAdi"] = sarkiAdi;
        dt.Rows.Add(row);
    }

    dataGridView1.DataSource = dt;
}
```

RASTGELE ŞARKI EKLEME

SQL Server veritabanındaki "Sarkilar" tablosundan rastgele bir şarkı adı çeken ve bu şarkıyı bir DataGridView kontrolünde gösteren bir Windows Forms uygulaması içerir.

RastgeleSarkiGetir fonksiyonu, SQL sorgusu kullanarak "Sarkilar" tablosundan rastgele bir şarkı adı alır ve bu şarkı adını geri döndürür. Ardından **btn_gününsarkisi_Click** olayı, bu rastgele şarkı adını alır ve bir DataTable oluşturarak DataGridView kontrolüne bu şarkı adını ekler, böylece kullanıcıya "Günün Şarkısı" olarak gösterilir. Hata durumlarında bir MessageBox ile hata mesajı gösterilir.

```
private string RastgeleSarkiGetir()
{
    try
    {
        baglan.Open();

        // Sarkilar tablosundan rastgele bir şarkı adı al
        string sorgu = "SELECT TOFl 1 SarkiAdi FROM Sarkilar ORDER
BY NEWID()";
        SqlCommand komut = new SqlCommand(sorgu, baglan);
        string rastgeleSarki = (string)komut.ExecuteScalar();

        return rastgeleSarki;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Hata: " + ex.Message);
        return null;
    }
    finally
    {
        baglan.Close();
    }
}

private void btn_gününsarkisi_Click(object sender, EventArgs e)
{
    string rastgeleSarki = RastgeleSarkiGetir();

    if (!string.IsNullOrEmpty(rastgeleSarki))
    {
        // DataGridView'de göstermek için yeni bir DataTable
        DataTable dt = new DataTable();
        dt.Columns.Add("Günün Şarkısı", typeof(string));
        dt.Rows.Add(rastgeleSarki);

        // DataGridView'e veriyi ata
        dataGridView3.DataSource = dt;
    }
}
```

YETMİŞLER.CS

SQL Server veritabanından 1970'li yıllara ait şarkı kayıtlarını içeren belirli bir döneme ait bilgileri çeker ve bu bilgileri bir DataGridView kontrolüne yükleyen bir Windows Forms uygulamasını temsil eder.

```
public void kayitlari_getir()
{
    baglan.Open();
    string getir = "SELECT
Sarkilar.SarkiAdi,Sarkilar.SarkiSuresi,turler.SarkiTuru AS
SarkiTuru,albumler.AlbumAdi AS AlbumAdi,sanaticilar.SanatciAdi AS
SanatciAdi FROM Sarkilar INNER JOIN turler ON Sarkilar.TurID =
turler.TurID INNER JOIN albumler ON Sarkilar.AlbumID = albumler.AlbumID
INNER JOIN sanaticilar ON Sarkilar.SanatciID = sanaticilar.SanatciID WHERE
Sarkilar.donemID = 1;";
    SqlCommand komut = new SqlCommand(getir, baglan);
    SqlDataAdapter ad = new SqlDataAdapter(komut); //listeleme
işlemi yapar datagridden

    DataTable dt = new DataTable();
    ad.Fill(dt); //datatable doldur
    dataGridView1.DataSource = dt;

    baglan.Close();
}
```

SEKSENLER.CS

SQL Server veritabanındaki "Sarkilar" tablosuyla ilişkili olan tur, album ve sanatçı bilgilerini içeren belirli bir döneme ait şarkıları çeker ve bu bilgileri bir DataGridView kontrolüne yükleyerek kullanıcıya gösterir.

```
public void kayitlari_getir()
{
    baglan.Open();
    string getir = "SELECT
Sarkilar.SarkiAdi,Sarkilar.SarkiSuresi,turler.SarkiTuru AS
SarkiTuru,albumler.AlbumAdi AS AlbumAdi,sanaticilar.SanatciAdi AS
SanatciAdi FROM Sarkilar INNER JOIN turler ON Sarkilar.TurID =
turler.TurID INNER JOIN albumler ON Sarkilar.AlbumID = albumler.AlbumID
INNER JOIN sanaticilar ON Sarkilar.SanatciID = sanaticilar.SanatciID WHERE
Sarkilar.donemID = 2;";
    SqlCommand komut = new SqlCommand(getir, baglan);
    SqlDataAdapter ad = new SqlDataAdapter(komut); //listeleme
işlemi yapar datagridden

    DataTable dt = new DataTable();
    ad.Fill(dt); //datatable doldur
    dataGridView1.DataSource = dt;

    baglan.Close();
}
```

DOKSANLAR.CS

SQL Server veritabanındaki "Sarkilar" tablosu ile ilişkili olan tür, albüm ve sanatçı bilgilerini içeren belirli bir döneme ait şarkı kayıtlarını çeker ve bu bilgileri bir DataGridView kontrolüne yükleyerek kullanıcıya gösterir. Şu anki sorgu, "Sarkilar" tablosundan "turler", "albumler" ve "sanatçılar" tablolarıyla birleştirilerek belirli bir döneme ait şarkıları getirmektedir.

```
public void kayitlari_getir()
{
    baglan.Open();
    string getir = "SELECT
Sarkilar.SarkiAdi,Sarkilar.SarkiSuresi,turler.SarkiTuru AS
SarkiTuru,albumler.AlbumAdi AS AlbumAdi,sanatçilar.SanatciAdi AS
SanatciAdi FROM Sarkilar INNER JOIN turler ON Sarkilar.TurID =
turler.TurID INNER JOIN albumler ON Sarkilar.AlbumID = albumler.AlbumID
INNER JOIN sanatçilar ON Sarkilar.SanatciID = sanatçilar.SanatciID WHERE
Sarkilar.donemID = 3;";
    SqlCommand komut = new SqlCommand(getir, baglan);
    SqlDataAdapter ad = new SqlDataAdapter(komut); //listeleme
işlemi yapar datagridden

    DataTable dt = new DataTable();
    ad.Fill(dt); //datatable doldur
    dataGridView1.DataSource = dt;

    baglan.Close();
}
```

İKİBİNLER.CS

SQL Server veritabanındaki "Sarkilar" tablosuyla ilişkili olan tür, albüm ve sanatçı bilgilerini içeren belirli bir döneme ait şarkıları çeker ve bu bilgileri bir DataGridView kontrolüne yükleyerek kullanıcıya gösterir.

```
public void kayitlari_getir()
{
    baglan.Open();
    string getir = "SELECT
Sarkilar.SarkiAdi,Sarkilar.SarkiSuresi,turler.SarkiTuru AS
SarkiTuru,albumler.AlbumAdi AS AlbumAdi,sanatçilar.SanatciAdi AS
SanatciAdi FROM Sarkilar INNER JOIN turler ON Sarkilar.TurID =
turler.TurID INNER JOIN albumler ON Sarkilar.AlbumID = albumler.AlbumID
INNER JOIN sanatçilar ON Sarkilar.SanatciID = sanatçilar.SanatciID WHERE
Sarkilar.donemID = 4;";
    SqlCommand komut = new SqlCommand(getir, baglan);
    SqlDataAdapter ad = new SqlDataAdapter(komut); //listeleme
işlemi yapar datagridden

    DataTable dt = new DataTable();
    ad.Fill(dt); //datatable doldur
    dataGridView1.DataSource = dt;

    baglan.Close();
}
```