**Bilgisayar Mühendisliği**

**Veritabanı 2 Dersi Final Projesi**

**Numara** : 211501047

**Ad Soyad** :Ömer Faruk Çetinkaya

**Projenin Konusu** : www.sikayetvar.com

**VİEW Soru ve Cevapları**

1-Kullanıcının tüm şikayetleriyle birlikte şikayet detaylarını ve firma bilgilerini içeren view:

CREATE VIEW vw\_KullaniciSikayetleri AS

SELECT K.KullaniciID, K.KullaniciAdi, S.SikayetID, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi,

F.FirmaID, F.Adi AS FirmaAdi, F.Adres, F.Telefon, F.Mail

FROM Kullanici K

JOIN Kullanici\_Sikayet KS ON K.KullaniciID = KS.KullaniciID

JOIN Sikayet S ON KS.SikayetID = S.SikayetID

JOIN Firma F ON S.FirmaID = F.FirmaID;

SELECT \* FROM vw\_KullaniciSikayetleri;

2-Şikayet yorumlarını listeleyen view:

CREATE VIEW vw\_SikayetYorumlari AS

SELECT sy.\*, k.KullaniciAdi

FROM Sikayet\_Yorumu sy

JOIN Kullanici k ON sy.KullaniciID = k.KullaniciID;

SELECT \* FROM vw\_SikayetYorumlari;

3-Bir kullanıcının yaptığı şikayetlerin sayısını içeren view:

CREATE VIEW vw\_KullaniciSikayetSayisi AS

SELECT k.KullaniciID, k.KullaniciAdi, COUNT(\*) AS SikayetSayisi

FROM Kullanici k

JOIN Kullanici\_Sikayet ks ON k.KullaniciID = ks.KullaniciID

GROUP BY k.KullaniciID, k.KullaniciAdi;

SELECT \* FROM vw\_KullaniciSikayetSayisi;

4-Bir kullanıcının tüm şikayetleri ve bu şikayetlere yapılan yorumları içeren view:

CREATE VIEW vw\_KullaniciSikayetYorumlari AS

SELECT K.KullaniciID, K.KullaniciAdi, S.SikayetID, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi,

SY.YorumID, SY.SikayetYorumu

FROM Kullanici K

JOIN Kullanici\_Sikayet KS ON K.KullaniciID = KS.KullaniciID

JOIN Sikayet S ON KS.SikayetID = S.SikayetID

LEFT JOIN Sikayet\_Yorumu SY ON K.KullaniciID = SY.KullaniciID AND KS.Kullanici\_Sikayet\_ID = SY.Kullanici\_Sikayet\_ID;

SELECT \* FROM vw\_KullaniciSikayetYorumlari;

5-KullaniciSikayetDetaylari

CREATE VIEW KullaniciSikayetDetaylari AS

SELECT Kullanici.KullaniciID, Kullanici.KullaniciAdi, Sikayet.Konu, Sikayet.Aciklama, Sikayet.YayinlanmaTarihi

FROM Kullanici

JOIN Kullanici\_Sikayet ON Kullanici.KullaniciID = Kullanici\_Sikayet.KullaniciID

JOIN Sikayet ON Kullanici\_Sikayet.SikayetID = Sikayet.SikayetID;

SELECT \* FROM KullaniciSikayetDetaylari;

6-FirmaCozumDetaylari

CREATE VIEW FirmaCozumDetaylari AS

SELECT Firma.FirmaID, Firma.Adi, Firma\_Cozum.Aciklama, Firma\_Cozum.YayinlanmaTarihi

FROM Firma

JOIN Firma\_Cozum ON Firma.FirmaID = Firma\_Cozum.FirmaID;

SELECT \* FROM FirmaCozumDetaylari;

7-firma sikayet sayılarını verir

CREATE VIEW FirmaSikayetSayilari AS

SELECT Firma.Adi, COUNT(Sikayet.SikayetID) AS SikayetSayisi

FROM Firma

LEFT JOIN Sikayet ON Firma.FirmaID = Sikayet.FirmaID

GROUP BY Firma.Adi;

SELECT \* FROM FirmaSikayetSayilari;

8-kullanıcının yaptığı yorumları listeler

CREATE VIEW KullaniciYapilanYorumlar AS

SELECT Kullanici.KullaniciAdi, Sikayet.Konu, Sikayet\_Yorumu.SikayetYorumu

FROM Kullanici

JOIN Sikayet\_Yorumu ON Kullanici.KullaniciID = Sikayet\_Yorumu.KullaniciID

JOIN Kullanici\_Sikayet ON Sikayet\_Yorumu.Kullanici\_Sikayet\_ID = Kullanici\_Sikayet.Kullanici\_Sikayet\_ID

JOIN Sikayet ON Kullanici\_Sikayet.SikayetID = Sikayet.SikayetID;

SELECT \* FROM KullaniciYapilanYorumlar;

9-Bir kullanıcının yaptığı şikayetlerin tarihlerini ve şikayet konularını içeren view:

CREATE VIEW vw\_KullaniciSikayetTarihleri AS

SELECT k.KullaniciAdi, s.Konu, s.YayinlanmaTarihi

FROM Kullanici k

JOIN Kullanici\_Sikayet ks ON k.KullaniciID = ks.KullaniciID

JOIN Sikayet s ON ks.SikayetID = s.SikayetID;

SELECT \* FROM vw\_KullaniciSikayetTarihleri;

10-Bir firma hakkındaki şikayetlerin tarihlerini ve açıklamalarını içeren view:

CREATE VIEW vw\_FirmaSikayetTarihleri AS

SELECT f.Adi, s.Aciklama, s.YayinlanmaTarihi

FROM Firma f

JOIN Sikayet s ON f.FirmaID = s.FirmaID;

SELECT \* FROM vw\_FirmaSikayetTarihleri;

**Storred Procedure Soru ve Cevapları**

1. Firma şikayetlerini beğeniye göre sıralar

CREATE PROCEDURE dbo.FirmaSikayetleriniBegeniyeGoreSirala

@FirmaID int

AS

BEGIN

SELECT S.SikayetID, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi

FROM Sikayet AS S

WHERE S.FirmaID = @FirmaID

ORDER BY S.BegeniSayisi DESC

END

EXEC dbo.FirmaSikayetleriniBegeniyeGoreSirala @FirmaID = 123;

1. En çok şikayet alan firmaları listeler

CREATE PROCEDURE dbo.EnCokSikayetAlanFirmalar

@Limit int

AS

BEGIN

SELECT TOP (@Limit) F.FirmaID, F.Adi, COUNT(S.SikayetID) AS SikayetSayisi

FROM Firma AS F

INNER JOIN Sikayet AS S ON F.FirmaID = S.FirmaID

GROUP BY F.FirmaID, F.Adi

ORDER BY SikayetSayisi DESC

END

EXEC dbo.EnCokSikayetAlanFirmalar @Limit = 5;

1. En popüler şikayetleri listeler

CREATE PROCEDURE dbo.EnPopulerSikayetler

@Limit int

AS

BEGIN

SELECT TOP (@Limit) S.SikayetID, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi

FROM Sikayet AS S

ORDER BY S.BegeniSayisi DESC

END

EXEC dbo.EnPopulerSikayetler @Limit = 10;

1. En yeni şikayetler

CREATE PROCEDURE dbo.EnYeniSikayetler

@Limit int

AS

BEGIN

SELECT TOP (@Limit) S.SikayetID, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi

FROM Sikayet AS S

ORDER BY S.YayinlanmaTarihi DESC

END

EXEC dbo.EnYeniSikayetler @Limit = 10;

1. Firma bilgilerini güncelleriz

CREATE PROCEDURE dbo.FirmaBilgiGuncelle

@FirmaID int,

@Adi nvarchar(50),

@Adres nvarchar(MAX),

@Telefon nvarchar(11),

@Mail nvarchar(50)

AS

BEGIN

UPDATE Firma

SET Adi = @Adi, Adres = @Adres, Telefon = @Telefon, Mail = @Mail

WHERE FirmaID = @FirmaID

END

EXEC dbo.FirmaBilgiGuncelle

@FirmaID = 123,

@Adi = 'Yeni Firma Adı',

@Adres = 'Yeni Firma Adresi',

@Telefon = 'Yeni Telefon Numarası',

@Mail = 'Yeni E-posta Adresi';

6-firma ekle

CREATE PROCEDURE dbo.FirmaEkle

@Adi nvarchar(50),

@Adres nvarchar(MAX),

@Telefon nvarchar(11),

@Mail nvarchar(50)

AS

BEGIN

INSERT INTO Firma (Adi, Adres, Telefon, Mail)

VALUES (@Adi, @Adres, @Telefon, @Mail)

END

EXEC dbo.FirmaEkle

@Adi = 'Yeni Firma',

@Adres = 'Yeni Firma Adresi',

@Telefon = 'Yeni Telefon Numarası',

@Mail = 'Yeni E-posta Adresi';

7-firma şikayet detayları

CREATE PROCEDURE dbo.FirmaSikayetDetay

@FirmaID int

AS

BEGIN

SELECT S.SikayetID, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi, F.Adi AS FirmaAdi

FROM Sikayet AS S

INNER JOIN Firma AS F ON S.FirmaID = F.FirmaID

WHERE S.FirmaID = @FirmaID

END

EXEC dbo.FirmaSikayetDetay @FirmaID = 123;

8-firma şikayet listele

CREATE PROCEDURE dbo.FirmaSikayetListele

@FirmaID int

AS

BEGIN

SELECT S.SikayetID, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi

FROM Sikayet AS S

WHERE S.FirmaID = @FirmaID

END

EXEC dbo.FirmaSikayetListele @FirmaID = 123;

9-kullanıcı şikayetleri

CREATE PROCEDURE dbo.KullaniciSikayetleri

@KullaniciID int

AS

BEGIN

SELECT S.SikayetID, S.FirmaID, F.Adi AS FirmaAdi, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi

FROM Sikayet AS S

INNER JOIN Firma AS F ON S.FirmaID = F.FirmaID

INNER JOIN Kullanici\_Sikayet AS KS ON S.SikayetID = KS.SikayetID

WHERE KS.KullaniciID = @KullaniciID

END

EXEC dbo.KullaniciSikayetleri @KullaniciID = 123;

10-kullanıcı ekle

CREATE PROCEDURE dbo.KullaniciEkle

@KullaniciAdi nvarchar(50),

@Parola nvarchar(50),

@Ad nvarchar(50),

@Soyad nvarchar(50),

@Mail nvarchar(50),

@Telefon nvarchar(11),

@Cinsiyet nvarchar(1)

AS

BEGIN

INSERT INTO Kullanici (KullaniciAdi, Parola, Ad, Soyad, Mail, Telefon, Cinsiyet)

VALUES (@KullaniciAdi, @Parola, @Ad, @Soyad, @Mail, @Telefon, @Cinsiyet)

END

EXEC dbo.KullaniciEkle

@KullaniciAdi = 'john.doe',

@Parola = 'password123',

@Ad = 'John',

@Soyad = 'Doe',

@Mail = 'john.doe@example.com',

@Telefon = '555-1234567',

@Cinsiyet = 'M'

11-kullanıcı şikayet listele

CREATE PROCEDURE dbo.KullaniciSikayetListele

@KullaniciID int

AS

BEGIN

SELECT S.SikayetID, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi, F.Adi AS FirmaAdi

FROM Sikayet AS S

INNER JOIN Kullanici\_Sikayet AS KS ON S.SikayetID = KS.SikayetID

INNER JOIN Firma AS F ON S.FirmaID = F.FirmaID

WHERE KS.KullaniciID = @KullaniciID

END

EXEC dbo.KullaniciSikayetListele @KullaniciID = 123

12-şikayet detay

CREATE PROCEDURE dbo.SikayetDetay

@SikayetID int

AS

BEGIN

SELECT S.SikayetID, S.FirmaID, F.Adi AS FirmaAdi, F.Adres, F.Telefon, F.Mail, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi

FROM Sikayet AS S

INNER JOIN Firma AS F ON S.FirmaID = F.FirmaID

WHERE S.SikayetID = @SikayetID

END

EXEC dbo.SikayetDetay @SikayetID = 123

13-şikayet filtrele

CREATE PROCEDURE dbo.FirmaSikayetFiltrele

@FirmaID int,

@MinBegeniSayisi int,

@MinYayinlanmaTarihi date

AS

BEGIN

SELECT S.SikayetID, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi

FROM Sikayet AS S

WHERE S.FirmaID = @FirmaID AND S.BegeniSayisi >= @MinBegeniSayisi AND S.YayinlanmaTarihi >= @MinYayinlanmaTarihi

END

EXEC dbo.FirmaSikayetFiltrele @FirmaID = 123, @MinBegeniSayisi = 10, @MinYayinlanmaTarihi = '2022-01-01'

14-şikayet yorumları ama yeni

CREATE PROCEDURE dbo.EnYeniSikayetYorumlariListele

AS

BEGIN

SELECT SY.YorumID, SY.SikayetYorumu, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi

FROM Sikayet\_Yorumu AS SY

INNER JOIN Kullanici\_Sikayet AS KS ON SY.Kullanici\_Sikayet\_ID = KS.Kullanici\_Sikayet\_ID

INNER JOIN Sikayet AS S ON KS.SikayetID = S.SikayetID

ORDER BY SY.YorumID DESC

END

EXEC dbo.EnYeniSikayetYorumlariListele

15-şikayet listele

CREATE PROCEDURE dbo.SikayetListele

AS

BEGIN

SELECT SikayetID, FirmaID, Konu, Aciklama, YayinlanmaTarihi, BegeniSayisi

FROM Sikayet

END

EXEC dbo.SikayetListele

16-şikayet oluştur

CREATE PROCEDURE dbo.SikayetOlustur

@FirmaID int,

@Konu nvarchar(100),

@Aciklama nvarchar(max),

@YayinlanmaTarihi date

AS

BEGIN

INSERT INTO Sikayet (FirmaID, Konu, Aciklama, YayinlanmaTarihi)

VALUES (@FirmaID, @Konu, @Aciklama, @YayinlanmaTarihi)

END

EXEC dbo.SikayetOlustur @FirmaID = 1, @Konu = 'Örnek Konu', @Aciklama = 'Örnek açıklama', @YayinlanmaTarihi = '2023-06-18'

17-şikayet ortalama beğeni sayısı

CREATE PROCEDURE dbo.FirmaSikayetOrtalamaBegeni

@FirmaID int,

@OrtalamaBegeni int OUTPUT

AS

BEGIN

SELECT @OrtalamaBegeni = AVG(BegeniSayisi)

FROM Sikayet

WHERE FirmaID = @FirmaID

END

DECLARE @OrtalamaBegeni int

EXEC dbo.FirmaSikayetOrtalamaBegeni @FirmaID = 1, @OrtalamaBegeni = @OrtalamaBegeni OUTPUT

SELECT @OrtalamaBegeni AS OrtalamaBegeniSayisi

18-şikayet yorumları

CREATE PROCEDURE dbo.SikayetYorumlari

@SikayetID int

AS

BEGIN

SELECT Y.YorumID, Y.KullaniciID, Y.Kullanici\_Sikayet\_ID, Y.SikayetYorumu

FROM Sikayet\_Yorumu AS Y

INNER JOIN Kullanici\_Sikayet AS KS ON Y.Kullanici\_Sikayet\_ID = KS.Kullanici\_Sikayet\_ID

WHERE KS.SikayetID = @SikayetID

END

EXEC dbo.SikayetYorumlari @SikayetID = 1

19-tarihe göre filtrele şikayetleri

CREATE PROCEDURE dbo.SikayetTarihFiltrele

@BaslangicTarihi date,

@BitisTarihi date

AS

BEGIN

SELECT S.SikayetID, S.Konu, S.Aciklama, S.YayinlanmaTarihi, S.BegeniSayisi, F.Adi AS FirmaAdi

FROM Sikayet AS S

INNER JOIN Firma AS F ON S.FirmaID = F.FirmaID

WHERE S.YayinlanmaTarihi BETWEEN @BaslangicTarihi AND @BitisTarihi

END

EXEC dbo.SikayetTarihFiltrele @BaslangicTarihi = '2023-01-01', @BitisTarihi = '2023-12-31'

20-

**Fonksiyon Soru ve Cevapları**

1. Şikayet çözümlerini listele

CREATE FUNCTION SikayetCozumleriniListele (

@SikayetID INT

)

RETURNS TABLE

AS

RETURN

SELECT fc.CozumID, f.Adi AS FirmaAdi, fc.Aciklama, fc.YayinlanmaTarihi

FROM Firma\_Cozum fc

INNER JOIN Firma f ON fc.FirmaID = f.FirmaID

WHERE fc.Kullanici\_Sikayet\_ID = @SikayetID

SELECT \*

FROM dbo.SikayetCozumleriniListele(@SikayetID)

1. Kullanıcı şikayetlerini listele

CREATE FUNCTION KullaniciSikayetleriniListele (

@KullaniciID INT

)

RETURNS TABLE

AS

RETURN

SELECT s.SikayetID, f.Adi AS FirmaAdi, s.Konu, s.Aciklama, s.YayinlanmaTarihi, s.BegeniSayisi

FROM Sikayet s

INNER JOIN Firma f ON s.FirmaID = f.FirmaID

INNER JOIN Kullanici\_Sikayet ks ON s.SikayetID = ks.SikayetID

WHERE ks.KullaniciID = @KullaniciID

SELECT \*

FROM dbo.KullaniciSikayetleriniListele(@KullaniciID)

1. Şikayetlerin ortalama beğeni sayısı

CREATE FUNCTION SikayetlerinOrtalamaBegeniSayisi ()

RETURNS INT

AS

BEGIN

DECLARE @OrtalamaBegeniSayisi INT

SELECT @OrtalamaBegeniSayisi = AVG(BegeniSayisi) FROM Sikayet

RETURN @OrtalamaBegeniSayisi

END

SELECT dbo.SikayetlerinOrtalamaBegeniSayisi()

4-şikayet yorumlarını listele

CREATE FUNCTION SikayetYorumlariniListele (

@SikayetID INT

)

RETURNS TABLE

AS

RETURN

SELECT sy.YorumID, k.KullaniciAdi, sy.SikayetYorumu

FROM Sikayet\_Yorumu sy

INNER JOIN Kullanici k ON sy.KullaniciID = k.KullaniciID

WHERE sy.Kullanici\_Sikayet\_ID = @SikayetID

SELECT \* FROM dbo.SikayetYorumlariniListele(1)

5-

**Trigger Soru ve Cevapları**

1. Firma silinmez

USE [sikayetcim] -- sikayetcim veritabanını kullanır

GO

/\*\*\*\*\*\* Nesne: Tetikleyici [dbo].[trg\_firmaSilinmez] Script Tarihi: 13.06.2023 11:01:37 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[trg\_firmaSilinmez] -- Tetikleyicinin adı

ON [dbo].[Firma] -- Tetikleyicinin uygulanacağı tablo

INSTEAD OF DELETE -- Silme işlemi yerine çalışacak

AS

BEGIN

RAISERROR('Firma Tablosu üzerinde kayıt silinmez', 1, 1) -- Hata mesajı döndürür

ROLLBACK TRANSACTION -- İşlemi geri alır

END

DELETE FROM Firma WHERE Id=1

2-cinsiyeti k olan silinmez

USE [sikayetcim] -- sikayetcim veritabanını kullanır

GO

/\*\*\*\*\*\* Nesne: Tetikleyici [dbo].[trg\_kizkullanicisilme] Script Tarihi: 13.06.2023 11:02:08 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[trg\_kizkullanicisilme] -- Tetikleyicinin adı

ON [dbo].[Kullanici] -- Tetikleyicinin uygulanacağı tablo

AFTER DELETE -- Silme işlemi sonrasında çalışacak

AS

BEGIN

IF(EXISTS(SELECT \* FROM deleted WHERE cinsiyet='K'))

BEGIN

RAISERROR('kız kullanıcıları silemezsiniz!!',0,0) -- Hata mesajı döndürür

ROLLBACK TRANSACTION -- İşlemi geri alır

END

END

DELETE FROM Kullanici WHERE Id=1

3-şikayet silindiği zaman şikayetler yedek tablosuna gönderir

USE [sikayetcim] -- sikayetcim veri tabanını kullanır

GO

/\*\*\*\*\*\* Nesne: Tetikleyici [dbo].[SikayetYedekler] Script Tarihi: 13.06.2023 11:02:34 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[SikayetYedekler] -- Tetikleyicinin adı

ON [dbo].[Sikayet] -- Tetikleyicinin uygulanacağı tablo

AFTER DELETE -- Silme işlemi sonrasında çalışacak

AS

BEGIN

INSERT INTO YedeklerSikayet -- 'YedeklerSikayet' tablosuna ekleme yapar

SELECT \* FROM deleted -- Silinen kayıtları seçer

END

DELETE FROM Sikayet WHERE Id=1

4-şikayet yorumu silindiği zaman bilgi gönderir

USE [sikayetcim] -- sikayetcim veri tabannı kullanır

GO

/\*\*\*\*\*\* Nesne: Tetikleyici [dbo].[silmeBilgisi] Script Tarihi: 13.06.2023 11:02:41 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[silmeBilgisi] -- Tetikleyicinin adı

ON [dbo].[Sikayet\_Yorumu] -- Tetikleyicinin uygulanacağı tablo

AFTER DELETE -- Silme işlemi sonrasında çalışacak

AS

BEGIN

SELECT 'sikayet yorumu Silindi' -- 'sikayet yorumu Silindi' metnini seçer

END

DELETE FROM Sikayet\_Yorumu WHERE Id=1

5-firma çözümüne veri eklendiğinde yeni tabloyu full getirir

USE [sikayetcim] -- sikayetcim veri tabanını kullanır

GO

/\*\*\*\*\*\* Nesne: Tetikleyici [dbo].[trg\_Listele] Script Tarihi: 13.06.2023 11:02:01 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[trg\_Listele] -- Tetikleyicinin adı

ON [dbo].[Firma\_Cozum] -- Tetikleyicinin uygulanacağı tablo

AFTER INSERT -- Ekleme işlemi sonrasında çalışacak

AS

BEGIN

SELECT \* FROM Firma\_Cozum -- 'Firma\_Cozum' tablosundaki tüm kayıtları seçer

END

**INSERT** **INTO** dbo.Firma\_Cozum (FirmaID, Aciklama, Kullanici\_Sikayet\_ID, YayinlanmaTarihi)

**SELECT** 2,'hemen ilgileniyoruz',5,'01.01.2023';