

Süperlig Otomasyonu

Proje Ekibindeki Kişiler:

- 205260050-Metehan TURGUT
- 205260006-Doğın SONAR
- 215260062-Serhan Burak YAŞAR

Dönem Projesi Gereksinimleri

Proje Süperligde bulunan takımların karşılaşmalarını, kupalarını stadlarını görebileceğimiz ve değiştirebileceğimiz şekilde tasarlanmıştır. Şu anda herhangi bir yekilendirilme ve yetkiye göre düzenleme yoktur. Yetkilendirme ve düzenleme işlemleri için gereken tasarımlar ve tablolar prosedürler oluşturulurken yeniden üzerinde durulacak dinamiklerdir:

Maclar:

Maçlar ayarlanabilecek silinebilecek ve düzenlenebilecek.

Futbolcu İstatistikleri:

Futbolcular istenilen bilgilere göre sıralanabilecek.

Lig Durumu:

Takımların maç sonuçlarının dökümü görünebilecek.

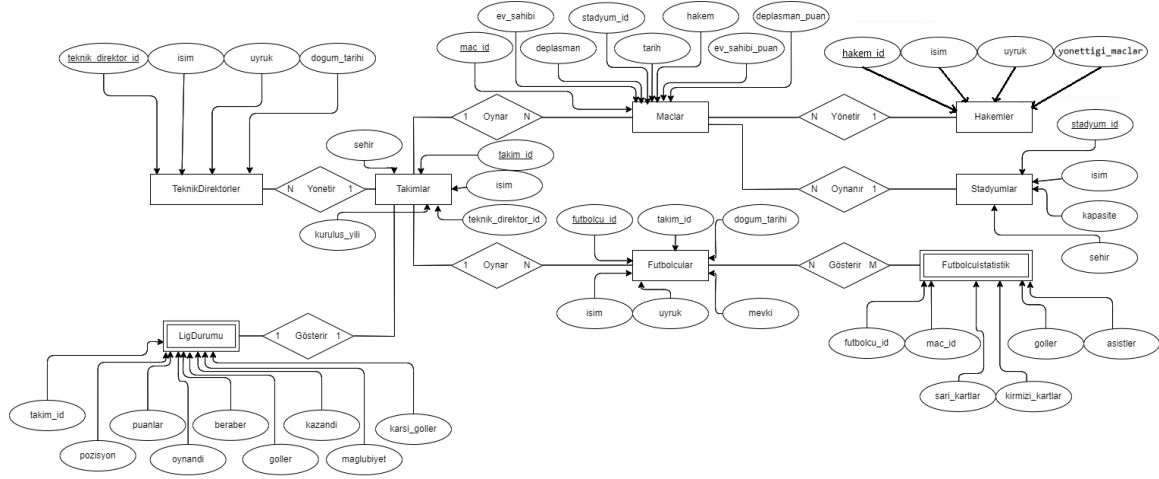
Varlıklar ve Nitelikleri

- Futbolcular (futbolcu_id, isim, dogum_tarihi, mevki, uyruk, takim_id)
- Takımlar (takim_id, isim, teknik_direktor_id, sehir, kurulus_yili)
- TeknikDirektorler (teknik_direktor_id, isim, uyruk, dogum_tarihi)
- Hakemler (hakem_id, isim, uyruk, yonettigi_maclar)
- Stadyumlar (stadyum_id, isim, sehir, kapasite)
- Maclar (mac_id, ev_sahibi, deplasman, tarih, stadyum_id, ev_sahibi_puan, deplasman_puan, hakem)
- FutbolcuIstatistik (futbolcu_id, mac_id, goller, asistler, sari_kartlar, kirmizi_kartlar)
- LigDurumu (takim_id, pozisyon, puanlar, oynandi, kazandi, beraber, maglubiyet, goller, karsi_goller)

İlişkiler

- Takımlar 1:N TeknikDirektorler
- Takımlar 1:N Maclar
- Takımlar 1:1 LigDurumu
- Takımlar 1:N Futbolcular
- Maclar N:1 Hakemler
- Maclar N:1 Stadyumlar
- Futbolcular N:M Futbolculstatistik

E-R Diagramı



Normalizasyon Süreci

1NF Ayırıştırma:

- Bütün kolonlar atomik olduğundan dolayı 1NF

2NF Ayırıştırma:

- Futbolculstatistik tablosunda her kolon birincil anahtara bağlı 2NF
- LigDurumu tablosunda her kolon doğrudan takım_id ile bağımlı kısmi bağımlılık yok 2NF

3NF Ayırıştırma:

- Takımlar tablosu Takımlar ve TakimBilgileri olarak iki ayrı tabloya ayırılır. Transitif bağımlılık ortadan kaldırılır. (isim -> şehir, kuruluş_yılı)
 - Takımlar (takım_id, teknik_direktor_id),
 - TakimBilgileri (takım_id, isim, şehir, kuruluş_yılı)

BCNF Ayırıştırma:

- Tüm tabloların her determinanı bir aday anahtarı olduđu için BCNF

Normalizasyon Sonrası İlişkiler:

- Takımlar 1:N TeknikDirektörler
- Takımlar 1:1 TakımBilgileri
- Takımlar 1:N Maclar
- Takımlar 1:1 LigDurumu
- Takımlar 1:N Futbolcular
- Maclar N:1 Hakemler
- Maclar N:1 Stadyumlar
- Futbolcular N:M Futbolcuİstatistik

SQL Kodları

```
CREATE DATABASE SuperligOtomasyonu;
```

```
USE SuperligOtomasyonu;
```

```
CREATE TABLE TeknikDirektorler (  
    teknik_direktor_id INT IDENTITY(1,1) PRIMARY KEY,  
    isim VARCHAR(255) NOT NULL,  
    uyruk VARCHAR(50),  
    dogum_tarihi DATE  
);
```

```
CREATE TABLE Takimler (  
    takim_id INT IDENTITY(1,1) PRIMARY KEY,  
    teknik_direktor_id INT,  
    FOREIGN KEY (teknik_direktor_id) REFERENCES TeknikDirektorler(teknik_direktor_id)  
);
```

```
CREATE TABLE TakimBilgileri (  
    takim_id INT IDENTITY(1,1) PRIMARY KEY,  
    isim VARCHAR(255) NOT NULL,  
    sehir VARCHAR(255),  
    kurulus_yili INT,  
    FOREIGN KEY (takim_id) REFERENCES Takimler(takim_id)  
);
```

```
CREATE TABLE Futbolcular (  
    futbolcu_id INT IDENTITY(1,1) PRIMARY KEY,  
    isim VARCHAR(255) NOT NULL,  
    soyisim VARCHAR(255) NOT NULL,  
    dogum_tarihi DATE,  
    pozisyon VARCHAR(50),  
    takım_id INT,  
    FOREIGN KEY (takım_id) REFERENCES Takimler(takım_id)  
);
```

```
futbolcu_id INT IDENTITY(1,1) PRIMARY KEY,  
takim_id INT,  
isim VARCHAR(255) NOT NULL,  
dogum_tarihi DATE,  
mevki VARCHAR(50),  
uyruk VARCHAR(50),  
FOREIGN KEY (takim_id) REFERENCES Takimler(takim_id)  
);
```

```
CREATE TABLE Hakemler (  
hakem_id INT IDENTITY(1,1) PRIMARY KEY,  
isim VARCHAR(255) NOT NULL,  
uyruk VARCHAR(50),  
yonettigi_maclar INT DEFAULT 0  
);
```

```
CREATE TABLE Stadyumlar (  
stadyum_id INT IDENTITY(1,1) PRIMARY KEY,  
isim VARCHAR(255) NOT NULL,  
sehir VARCHAR(255),  
kapasite INT  
);
```

```
CREATE TABLE Maclar (  
mac_id INT IDENTITY(1,1) PRIMARY KEY,  
ev_sahibi INT,  
deplasman INT,  
tarih DATETIME,
```

```
stadyum_id INT,  
ev_sahibi_puan INT,  
deplasman_puan INT,  
hakem INT,  
FOREIGN KEY (ev_sahibi) REFERENCES Takimler(takim_id),  
FOREIGN KEY (deplasman) REFERENCES Takimler(takim_id),  
FOREIGN KEY (stadyum_id) REFERENCES Stadyumlar(stadyum_id),  
FOREIGN KEY (hakem) REFERENCES Hakemler(hakem_id)  
);
```

```
CREATE TABLE FutbolcuIstatistik (  
    futbolcu_id INT,  
    mac_id INT,  
    goller INT DEFAULT 0,  
    asistler INT DEFAULT 0,  
    sari_kartlar INT DEFAULT 0,  
    kirmizi_kartlar INT DEFAULT 0,  
    PRIMARY KEY (futbolcu_id, mac_id),  
    FOREIGN KEY (futbolcu_id) REFERENCES Futbolcular(futbolcu_id),  
    FOREIGN KEY (mac_id) REFERENCES Maclar(mac_id)  
);
```

```
CREATE TABLE LigDurumu (  
    takim_id INT,  
    pozisyon INT,  
    puanlar INT DEFAULT 0,  
    oynandi INT DEFAULT 0,  
    kazandi INT DEFAULT 0,
```

```
beraber INT DEFAULT 0,  
maglubiyet INT DEFAULT 0,  
goller INT DEFAULT 0,  
karsi_goller INT DEFAULT 0,  
PRIMARY KEY (takim_id),  
FOREIGN KEY (takim_id) REFERENCES Takimler(takim_id)  
);
```

Teknik direktörlerin doğum tarihlerini kontrol eden trigger

```
CREATE TRIGGER CheckDogumTarihi  
ON TeknikDirektorler  
FOR INSERT, UPDATE  
AS  
BEGIN  
    DECLARE @dogum_tarihi DATE;  
    SELECT @dogum_tarihi = dogum_tarihi FROM inserted;  
  
    IF @dogum_tarihi > GETDATE()  
    BEGIN  
        RAISERROR('Doğum tarihi gelecekte olamaz.', 16, 1);  
        ROLLBACK TRANSACTION;  
    END  
END;
```

Takımların teknik direktörlerinin olup olmadığını kontrol eden trigger

CREATE TRIGGER CheckTeknikDirektor

ON Takimler

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @teknik_direktor_id INT;

SELECT @teknik_direktor_id = teknik_direktor_id FROM inserted;

IF NOT EXISTS (SELECT 1 FROM TeknikDirektorler WHERE teknik_direktor_id = @teknik_direktor_id)

BEGIN

RAISERROR('Takım için geçerli bir teknik direktör bulunmuyor.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

Futbolcuların yaşının 16'dan büyük olup olmadığını kontrol eden trigger

```
CREATE TRIGGER CheckFutbolcuYasi
ON Futbolcular
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @dogum_tarihi DATE;
    DECLARE @yas INT;
    SELECT @dogum_tarihi = dogum_tarihi FROM inserted;

    SET @yas = DATEDIFF(YEAR, @dogum_tarihi, GETDATE());
    IF @yas < 16
    BEGIN
        RAISERROR('Futbolcu 16 yaşından küçük olamaz.', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
```

Maçlardan sonra çeşitli güncellemeler yapan trigger

```
CREATE TRIGGER UpdateFutbolcuIstatistik
```

```
ON Maclar
```

```
FOR UPDATE
```

```
AS
```

```
BEGIN
```

```
    DECLARE @mac_id INT;
```

```
    DECLARE @ev_sahibi_puan INT;
```

```
    DECLARE @deplasman_puan INT;
```

```
    SELECT @mac_id = mac_id, @ev_sahibi_puan = ev_sahibi_puan, @deplasman_puan =  
    deplasman_puan
```

```
    FROM inserted;
```

```
    UPDATE FutbolcuIstatistik
```

```
    SET goller = goller + (SELECT COUNT(*) FROM Futbolcular
```

```
        WHERE takim_id = (SELECT ev_sahibi FROM Maclar WHERE mac_id = @mac_id)
```

```
        AND futbolcu_id IN (SELECT futbolcu_id FROM FutbolcuIstatistik WHERE mac_id =  
@mac_id AND goller > 0)),
```

```
    asistler = asistler + (SELECT COUNT(*) FROM Futbolcular
```

```
        WHERE takim_id = (SELECT ev_sahibi FROM Maclar WHERE mac_id = @mac_id)
```

```
        AND futbolcu_id IN (SELECT futbolcu_id FROM FutbolcuIstatistik WHERE mac_id =  
@mac_id AND asistler > 0))
```

```
    WHERE mac_id = @mac_id;
```

```
    UPDATE FutbolcuIstatistik
```

```
    SET goller = goller + (SELECT COUNT(*) FROM Futbolcular
```

```
        WHERE takim_id = (SELECT deplasman FROM Maclar WHERE mac_id = @mac_id)
```

AND futbolcu_id IN (SELECT futbolcu_id FROM FutbolcuIstatistik WHERE mac_id = @mac_id AND goller > 0)),

asistler = asistler + (SELECT COUNT(*) FROM Futbolcular

WHERE takim_id = (SELECT deplasman FROM Maclar WHERE mac_id = @mac_id)

AND futbolcu_id IN (SELECT futbolcu_id FROM FutbolcuIstatistik WHERE mac_id = @mac_id AND asistler > 0))

WHERE mac_id = @mac_id;

UPDATE FutbolcuIstatistik

SET sari_kartlar = sari_kartlar + (SELECT COUNT(*) FROM Futbolcular

WHERE takim_id = (SELECT ev_sahibi FROM Maclar WHERE mac_id = @mac_id)

AND futbolcu_id IN (SELECT futbolcu_id FROM FutbolcuIstatistik WHERE mac_id = @mac_id AND sari_kartlar > 0))

WHERE mac_id = @mac_id;

UPDATE FutbolcuIstatistik

SET sari_kartlar = sari_kartlar + (SELECT COUNT(*) FROM Futbolcular

WHERE takim_id = (SELECT deplasman FROM Maclar WHERE mac_id = @mac_id)

AND futbolcu_id IN (SELECT futbolcu_id FROM FutbolcuIstatistik WHERE mac_id = @mac_id AND sari_kartlar > 0))

WHERE mac_id = @mac_id;

UPDATE FutbolcuIstatistik

SET kirmizi_kartlar = kirmizi_kartlar + (SELECT COUNT(*) FROM Futbolcular

WHERE takim_id = (SELECT ev_sahibi FROM Maclar WHERE mac_id = @mac_id)

AND futbolcu_id IN (SELECT futbolcu_id FROM FutbolcuIstatistik WHERE mac_id = @mac_id AND kirmizi_kartlar > 0))

WHERE mac_id = @mac_id;

```
UPDATE Futbolculstatistik

SET kirmizi_kartlar = kirmizi_kartlar + (SELECT COUNT(*) FROM Futbolcular

WHERE takim_id = (SELECT deplasman FROM Maclar WHERE mac_id =

@mac_id)

AND futbolcu_id IN (SELECT futbolcu_id FROM Futbolculstatistik WHERE

mac_id = @mac_id AND kirmizi_kartlar > 0))

WHERE mac_id = @mac_id;

END;
```

Takımların kuruluş yılını kontrol etmek için constraint

```
ALTER TABLE TakimBilgileri

ADD CONSTRAINT CHK_KurulusYili CHECK (kurulus_yili > 1800 AND kurulus_yili <=

YEAR(GETDATE()));
```

Futbolcuların mevkisini kontrol etmek için constraint

```
ALTER TABLE Futbolcular

ADD CONSTRAINT CHK_Mevki CHECK (mevki IN ('Forvet', 'Orta Saha', 'Defans', 'Kaleci'));
```

Hakemlerin yönettiği maç sayısının negatif olup olmadığını kontrol etmek için constraint

```
ALTER TABLE Hakemler

ADD CONSTRAINT CHK_YonettigiMaclar CHECK (yonettigi_maclar >= 0);
```

Stadyumların kapasitesini kontrol etmek için constraint

```
ALTER TABLE Stadyumlar

ADD CONSTRAINT CHK_Kapasite CHECK (kapasite > 0);
```

Takımların puanlarının negatif olup olmadığını kontrol etmek için constraint

```
ALTER TABLE Maclar

ADD CONSTRAINT CHK_Puanlar CHECK (ev_sahibi_puan >= 0 AND deplasman_puan >= 0);
```

Futbolcuların asist ve gol sayısının negatif olup olmadığını kontrol etmek için constraint

ALTER TABLE FutbolculStatistik

ADD CONSTRAINT CHK_Goller CHECK (goller >= 0 AND asistler >= 0);

Lig Durumunu kontrol etmek için constraint

ALTER TABLE LigDurumu

ADD CONSTRAINT CHK_LigDurumu CHECK (puanlar >= 0 AND oynandi >= 0 AND kazandi >= 0 AND beraber >= 0 AND maglubiyet >= 0);

Teknik Direktör Ekleyen Prosedür

CREATE PROCEDURE AddTeknikDirektor

@isim VARCHAR(255),

@uyruk VARCHAR(50),

@dogum_tarihi DATE

AS

BEGIN

INSERT INTO TeknikDirektorler (isim, uyruk, dogum_tarihi)

VALUES (@isim, @uyruk, @dogum_tarihi);

END;

Takım Ekleyen Prosedür

```
CREATE PROCEDURE AddTakim
```

```
    @teknik_direktor_id INT,
```

```
    @isim VARCHAR(255)
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO Takimlar (teknik_direktor_id)
```

```
    VALUES (@teknik_direktor_id);
```

```
    INSERT INTO TakimBilgileri (isim)
```

```
    VALUES (@isim);
```

```
END;
```

Futbolcu Ekleyen Prosedür

```
CREATE PROCEDURE AddFutbolcu
```

```
    @takim_id INT,
```

```
    @isim VARCHAR(255),
```

```
    @dogum_tarihi DATE,
```

```
    @mevki VARCHAR(50),
```

```
    @uyruk VARCHAR(50)
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO Futbolcular (takim_id, isim, dogum_tarihi, mevki, uyruk)
```

```
    VALUES (@takim_id, @isim, @dogum_tarihi, @mevki, @uyruk);
```

```
END;
```

Futbolcu İstatistiği Ekleyen Prosedür

```
CREATE PROCEDURE AddFutbolcuIstatistik
```

```
    @futbolcu_id INT,
```

```
    @mac_id INT,
```

```
    @goller INT = 0,
```

```
    @asistler INT = 0,
```

```
    @sari_kartlar INT = 0,
```

```
    @kirmizi_kartlar INT = 0
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO FutbolcuIstatistik (futbolcu_id, mac_id, goller, asistler, sari_kartlar, kirmizi_kartlar)
```

```
    VALUES (@futbolcu_id, @mac_id, @goller, @asistler, @sari_kartlar, @kirmizi_kartlar);
```

```
END;
```

Hakem Ekleyen Prosedür

```
CREATE PROCEDURE AddHakem
```

```
    @isim VARCHAR(255),
```

```
    @uyruk VARCHAR(50)
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO Hakemler (isim, uyruk)
```

```
    VALUES (@isim, @uyruk);
```

```
END;
```

Stadyum Ekleyen Prosedür

```
CREATE PROCEDURE AddStadyum
    @isim VARCHAR(255),
    @sehir VARCHAR(255),
    @kapasite INT
AS
BEGIN
    INSERT INTO Stadyumlar (isim, sehir, kapasite)
    VALUES (@isim, @sehir, @kapasite);
END;
```

Maç Ekleyen Prosedür

```
CREATE PROCEDURE AddMac
    @ev_sahibi INT,
    @deplasman INT,
    @tarih DATETIME,
    @stadyum_id INT,
    @ev_sahibi_puan INT,
    @deplasman_puan INT,
    @hakem INT
AS
BEGIN
    INSERT INTO Maclar (ev_sahibi, deplasman, tarih, stadyum_id, ev_sahibi_puan, deplasman_puan,
hakem)
    VALUES (@ev_sahibi, @deplasman, @tarih, @stadyum_id, @ev_sahibi_puan, @deplasman_puan,
@hakem);
END;
```


Lig Durumunu Güncelleyen Prosedür

CREATE PROCEDURE UpdateLigDurumu

@takim_id INT,

@pozisyon INT,

@puanlar INT,

@oynandi INT,

@kazandi INT,

@beraber INT,

@maglubiyet INT,

@goller INT,

@karsi_goller INT

AS

BEGIN

UPDATE LigDurumu

SET pozisyon = @pozisyon,

puanlar = @puanlar,

oynandi = @oynandi,

kazandi = @kazandi,

beraber = @beraber,

maglubiyet = @maglubiyet,

goller = @goller,

karsi_goller = @karsi_goller

WHERE takim_id = @takim_id;

END;

Maç Sonucunu Güncelleyen Prosedür

```
CREATE PROCEDURE UpdateMacSonucu
```

```
    @mac_id INT,
```

```
    @ev_sahibi_puan INT,
```

```
    @deplasman_puan INT
```

```
AS
```

```
BEGIN
```

```
    UPDATE Maclar
```

```
    SET ev_sahibi_puan = @ev_sahibi_puan,
```

```
        deplasman_puan = @deplasman_puan
```

```
    WHERE mac_id = @mac_id;
```

```
    DECLARE @ev_sahibi INT, @deplasman INT;
```

```
    SELECT @ev_sahibi = ev_sahibi, @deplasman = deplasman FROM Maclar WHERE mac_id = @mac_id;
```

```
    UPDATE LigDurumu
```

```
    SET puanlar = puanlar + @ev_sahibi_puan
```

```
    WHERE takim_id = @ev_sahibi;
```

```
    UPDATE LigDurumu
```

```
    SET puanlar = puanlar + @deplasman_puan
```

```
    WHERE takim_id = @deplasman;
```

```
END;
```

Test Sorguları ile kodların çalışıp çalışmadığını kontrol edebilirsiniz

use SuperligOtomasyonu;

EXEC AddTeknikDirektor 'Fatih Terim', 'Türkiye', '1953-09-04';

EXEC AddTeknikDirektor 'Pep Guardiola', 'İspanya', '1971-01-18';

EXEC AddTeknikDirektor 'Jürgen Klopp', 'Almanya', '2025-13-01';

EXEC AddTakim 1, 'Galatasaray';

EXEC AddTakim 2, 'Barcelona';

EXEC AddTakim 999, 'Hatalı Takım';

EXEC AddFutbolcu 1, 'Lionel Messi', '1987-06-24', 'Forvet', 'Arjantin';

EXEC AddFutbolcu 2, 'Gerard Pique', '1987-02-02', 'Defans', 'İspanya';

EXEC AddFutbolcu 999, 'Cristiano Ronaldo', '1985-02-05', 'Forvet', 'Portekiz';

EXEC AddHakem 'Cüneyt Çakır', 'Türkiye';

EXEC AddHakem 'Bjorn Kuipers', 'Hollanda';

EXEC AddHakem NULL, 'Türkiye';

EXEC AddStadyum 'Camp Nou', 'Barcelona', 99354;

EXEC AddStadyum 'Türk Telekom Arena', 'İstanbul', 52650;

EXEC AddStadyum 'Hatalı Stadyum', 'İstanbul', -5000;

EXEC AddMac 1, 2, '2025-01-15 20:00:00', 1, 3, 1, 1;

EXEC AddMac 999, 2, '2025-01-15 20:00:00', 1, 3, 1, 1;

EXEC AddFutbolcuIstatistik 1, 1, 2, 1, 0, 0;

EXEC AddFutbolcuIstatistik 2, 1, 0, 0, 1, 0;

EXEC AddFutbolcuIstatistik 999, 1, 1, 0, 0, 0;

EXEC UpdateLigDurumu 1, 1, 3, 1, 1, 0, 0, 2, 0;

EXEC UpdateLigDurumu 2, 2, 1, 1, 0, 1, 0, 1, 2;

EXEC UpdateLigDurumu 999, 3, 0, 1, 0, 0, 1, 0, 2;