



Mühendislik Fakültesi Bilgisayar Mühendisliği

BMÜ329 VERİ TABANI SİSTEMLERİ

Proje Adı: Futbol Scout (Yetenek avcısı) Sistemi

Proje Ekibi:

- 225260044 Furkan Çam
- 225260040 Yunus Emre İnel
- 225260136 Davut Karagöl

1.Giriş: Projenin Neden Gerekli Olduğu:

Scout ekiplerinin karşılaştığı temel problemleri listeleterek bu sistemin nasıl bir çözüm sunduğunu anlatabilirsiniz. Örneğin:

- Dağınık veri yönetimi
- Veri erişiminde yaşanan zorluklar
- Manüel analiz ve raporlama süreçlerinin zaman alıcı olması

2. Hedef Kitleyi Tanımlama

- Sistem kimler için geliştiriliyor?
 - Scout ekipleri
 - Teknik direktörler

- Kulüp yönetimleri
 - Veri analistleri
-

3. Proje Özeti:

"Futbol Sektöründe Oyuncu ve Performans Takip Sistemi" projesi, futbol sektöründeki oyuncu izleme ve değerlendirme süreçlerini modernize etmeyi amaçlayan bir projedir. Scout ekiplerinin oyuncu performansını takip edebileceği, transfer geçmişlerini analiz edebileceği ve detaylı raporlar oluşturabileceği bir sistem sunulmaktadır.

Projenin temel amacı, futbol takımları ve scout ekipleri için hızlı, güvenilir ve detaylı bir bilgi yönetim platformu sağlamaktır. Bununla birlikte, sistem üzerinden alınan analizlerin, karar alma süreçlerine etkin bir katkı sağlaması hedeflenmektedir.

4. Proje Ayrıntıları:

4.1. Proje Amaçları:

- Oyuncu performans takibini kolaylaştırmak ve verilerin düzenli bir şekilde saklanmasını sağlamak.
- Scout raporlarını güvenli ve erişilebilir bir platformda yönetmek.
- Transfer geçmişlerini analiz ederek stratejik karar alma süreçlerini desteklemek.
- Kullanıcı dostu bir arayüzle tüm süreçlerin hızlı ve etkili bir şekilde yönetilmesini sağlamak.

4.2. Proje Süreci:

- Oyuncu bilgilerinin saklanacağı bir veri tabanı oluşturulması ve ilişkisel veri modellerinin tasarlanması.
- Performans ve transfer bilgilerini kaydedecek modüllerin geliştirilmesi.
- Scout raporlarının sisteme entegre edilmesi ve kolay erişim sağlanması.
- Geliştirilen sistemin test edilmesi ve geri bildirimlere göre optimize edilmesi.

4.3. Kullanılan Teknolojiler:

- **SQL Server Management Studio 2022:**
Veri tabanı tasarımı, veri yönetimi ve ilişkisel veri modellerinin oluşturulması için kullanılacaktır.
-

5. Proje Sonuçları:

Bu proje tamamlandığında, futbol sektörü için aşağıdaki avantajlar sağlanacaktır:

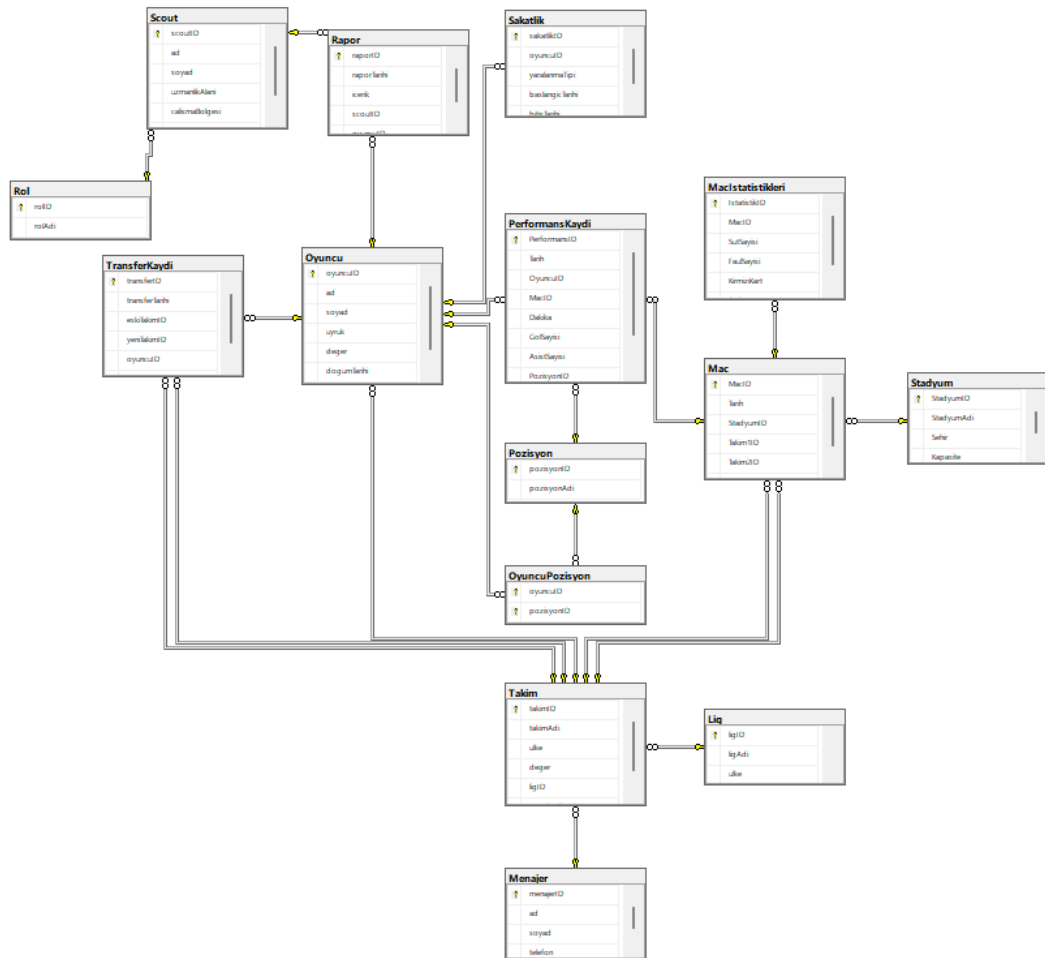
- Oyuncu performansı ve transfer bilgilerinin etkin bir şekilde yönetilmesi.
- Scout ekipleri için detaylı raporlama ve analiz araçları sunulması.
- Sistematik bir veri tabanı altyapısıyla bilgilerin güvenli bir şekilde saklanması.
- Karar alma süreçlerinin hızlanması ve veri doğruluğunun artırılması.

6. Sonuç:

"Futbol Sektöründe Oyuncu ve Performans Takip Sistemi" projesi, geleneksel izleme yöntemlerinden modern, veri tabanı tabanlı bir yapıya geçişi destekleyen kapsamlı bir çözümdür. Proje ekibi, süreç boyunca iş birliği içinde çalışarak hedeflenen amaçlara ulaşmayı planlamaktadır. Bu proje, futbol sektörü için yalnızca bir yazılım çözümü değil, aynı zamanda daha etkin ve verimli bir işleyiş sağlayan bir inovasyon aracıdır.

Elde edilen deneyimler, benzer projelerin geliştirilmesi için rehberlik edecektir.

7.UML DİAGRAMI:



8.Varlık-Özellik Tablosu:

Varlık-Özellik Tablosu:

Varlıklar	Özellikler
Oyuncu	Oyuncu ID, Ad, Soyad, Uyrak, Değer, Doğum Tarihi
Takım	Takım ID, Takım Adı, Değer, Ülke, Lig ID, Menajer ID
Lig	Lig ID, Lig Adı, Ülke
Pozisyon	Pozisyon ID, Pozisyon Adı
Rol	Rol ID, Rol Adı
Transfer Kaydı	Transfer ID, Transfer Tarihi, Eski Takım ID, Yeni Takım ID, Oyuncu ID, Transfer Ücreti
Menajer	Menajer ID, Ad, Soyad, Telefon
Sakatlık	Sakatlık ID, Oyuncu ID, Yaralanma Tipi, Başlangıç Tarihi, Bitiş Tarihi
Rapor	Rapor ID, Rapor Tarihi, İçerik, Scout ID, Oyuncu ID
Maç İstatistikleri	İstatistik ID, Maç ID, Şut Sayısı, Faul Sayısı, Kırmızı Kart, Sarı Kart
Stadyum	Stadyum ID, Stadyum Ad, Şehir, Kapasite
Maç	Maç ID, Tarih, Stadyum ID, Takım1 ID, Takım2 ID, Sonuç
Performans Kaydı	Performans ID, Tarih, Oyuncu ID, Maç ID, Dakika, Gol Sayısı, Asist Sayısı, Pozisyon, Scout Yorum
Scout	Scout ID, Ad, Soyad, Uzmanlık Alanı, Çalışma Bölgesi, Rol ID

9.Varlık-İlişki Tablosu:

Varlık İlişki Tablosu

Bu tablo, veritabanındaki varlıklar arasındaki ilişkileri göstermektedir:

İlişki	Açıklama
Oyuncu - Performans Kaydı (1-N)	Bir oyuncu, birden fazla performans kaydına sahip olabilir. Ancak her performans kaydı yalnızca bir oyuncuya aittir.
Oyuncu - Rapor (1-N)	Bir oyuncu hakkında birden fazla rapor oluşturulabilir. Ancak her rapor yalnızca bir oyuncuya aittir.
Oyuncu - Takım (N-1)	Bir oyuncu aynı anda sadece bir takımda olabilir. Fakat bir takımda birden fazla oyuncu bulunabilir.
Oyuncu - Transfer Kaydı (1-N)	Bir oyuncunun birden fazla transfer kaydı olabilir. Ancak her transfer kaydı yalnızca bir oyuncuya aittir.
Oyuncu - Sakatlık (1-N)	Bir oyuncunun birden fazla sakatlık kaydı olabilir. Her sakatlık kaydı yalnızca bir oyuncuya aittir.
Oyuncu - Pozisyon (N-M)	Bir oyuncu birden fazla pozisyonda oynayabilir. Aynı zamanda bir pozisyonda birden fazla oyuncu bulunabilir. Bu ilişki çift yönlü çoktan-çoka (N-M) bir ilişkidir.
Scout - Performans Kaydı (1-N)	Bir scout birden fazla performans kaydı oluşturabilir. Ancak her performans kaydı yalnızca bir scout tarafından oluşturulabilir.
Scout - Rapor (1-N)	Bir scout, birden fazla rapor oluşturabilir. Ancak her rapor yalnızca bir scout'a aittir.
Scout - Rol (N-1)	Birden fazla scout aynı role sahip olabilir. Ancak her scout yalnızca bir role sahip olabilir.
Takım - Transfer Kaydı (1-N)	Bir takımın birden fazla transfer kaydı olabilir. Ancak her transfer kaydı yalnızca bir takıma aittir.
Takım - Lig (N-1)	Birden fazla takım aynı ligde olabilir. Ancak her takım yalnızca bir ligde yer alır.
Takım - Menajer (N-1)	Birden fazla takım aynı menajer tarafından yönetilebilir. Ancak her takım yalnızca bir menajer tarafından yönetilir.
Maç - Performans Kaydı (N-1)	Bir maçta birden fazla performans kaydı olabilir. Ancak her performans kaydı yalnızca bir maç için geçerlidir.
Maç - Stadyum (N-1)	Birden fazla maç aynı stadyumda oynanabilir. Ancak her maç yalnızca bir stadyumda yapılır.
Maç - Maç İstatistikleri (1-1)	Her maçın yalnızca bir istatistik kaydı vardır ve her istatistik kaydı yalnızca bir maça aittir. Bu, bire bir (1-1) bir ilişki türüdür.

10.Sql kodları:

10.1. Oluşturacağımız tablolar ve eklenen örnek veriler:

-- Lig Tablosu --

```
CREATE TABLE Lig (  
    ligID INT IDENTITY(1,1) PRIMARY KEY,  
    ligAdi VARCHAR(50),  
    ulke VARCHAR(50)  
);  
  
INSERT INTO Lig (ligAdi, ulke)  
VALUES  
('Süper Lig', 'Türkiye'),  
('Premier League', 'İngiltere'),  
('Serie A', 'İtalya'),  
('La Liga', 'İspanya'),  
('Bundesliga', 'Almanya');
```

-- Pozisyon Tablosu -----

```
CREATE TABLE Pozisyon (  
    pozisyonID INT IDENTITY(1,1) PRIMARY KEY,  
    pozisyonAdi VARCHAR(50)  
);  
  
INSERT INTO Pozisyon (pozisyonAdi)  
VALUES  
('Kaleci'),  
('Defans'),  
('Orta Saha'),  
('Forvet'),  
('Kanat');
```

-- Menajer Tablosu --

```
CREATE TABLE Menajer (  
    menajerID INT IDENTITY(1,1) PRIMARY KEY,  
    ad VARCHAR(50),  
    soyad VARCHAR(50),  
    telefon VARCHAR(20)  
);
```

```
INSERT INTO Menajer (ad, soyad, telefon)  
VALUES  
('Ahmet', 'Yilmaz', '123456789'),  
('James', 'Smith', '987654321'),  
('Marco', 'Rossi', '456789123'),  
('Miguel', 'Lopez', '789123456'),  
('Klaus', 'Muller', '321654987');
```

-- Takim Tablosu --

```
CREATE TABLE Takim (  
    takimID INT IDENTITY(1,1) PRIMARY KEY,  
    takimAdi VARCHAR(50),  
    ulke VARCHAR(50),  
    deger DECIMAL(20,2),  
    ligID INT,  
    menajerID INT  
    FOREIGN KEY (menajerID) REFERENCES Menajer(menajerID),  
    FOREIGN KEY (ligID) REFERENCES Lig(ligID)  
);
```

INSERT INTO Takim (takimAdi, ulke, deger, ligID, menajerID)

VALUES

('Galatasaray', 'Türkiye', 20000000, 1, 1),

('Manchester United', 'İngiltere', 60000000, 2, 2),

('Juventus', 'İtalya', 45000000, 3, 3),

('Real Madrid', 'İspanya', 80000000, 4, 4),

('Bayern Munich', 'Almanya', 70000000, 5, 5);

-- Oyuncu Tablosu -----

CREATE TABLE Oyuncu (

oyuncuID INT IDENTITY(1,1) PRIMARY KEY,

ad VARCHAR(50),

soyad VARCHAR(50),

uyruk VARCHAR(50),

deger DECIMAL(20,2),

dogumTarihi DATE,

takimID INT,

FOREIGN KEY (takimID) REFERENCES Takim(takimID)

);

INSERT INTO Oyuncu (ad, soyad, uyruk, deger, dogumTarihi, takimID)

VALUES

('Mehmet', 'Kaya', 'Türkiye', 500000, '2000-05-15', 1),

('John', 'Doe', 'İngiltere', 300000, '1999-07-20', 2),

('Carlos', 'Santana', 'Brezilya', 600000, '2001-03-10', 3),

('Ali', 'Demir', 'Türkiye', 250000, '1998-11-02', 4),

('Sven', 'Larsson', 'İsveç', 400000, '2002-01-18', 5);

-- Oyuncu-Pozisyon Tablosu --

```
CREATE TABLE OyuncuPozisyon (  
    oyuncuID INT,  
    pozisyonID INT,  
    PRIMARY KEY (oyuncuID, pozisyonID),  
    FOREIGN KEY (oyuncuID) REFERENCES Oyuncu(oyuncuID),  
    FOREIGN KEY (pozisyonID) REFERENCES Pozisyon(pozisyonID)  
);
```

```
INSERT INTO OyuncuPozisyon (oyuncuID, pozisyonID) VALUES  
(1, 4),  
(2, 3),  
(3, 5),  
(4, 2),  
(5, 1);
```

-- Transfer Kaydi Tablosu --

```
CREATE TABLE TransferKaydi (  
    transferID INT IDENTITY(1,1) PRIMARY KEY,  
    transferTarihi DATE,  
    eskiTakimID INT,  
    yeniTakimID INT,  
    oyuncuID INT,  
    transferUcreti DECIMAL(15, 2),  
    FOREIGN KEY (eskiTakimID) REFERENCES Takim(takimID),  
    FOREIGN KEY (yeniTakimID) REFERENCES Takim(takimID),  
    FOREIGN KEY (oyuncuID) REFERENCES Oyuncu(oyuncuID)  
);
```

INSERT INTO TransferKaydi (transferTarihi, eskiTakimID, yeniTakimID, oyuncuID, transferUcreti)

VALUES

('2023-01-15', 1, 2, 1, 100000),

('2023-06-20', 3, 4, 2, 200000),

('2022-12-10', 2, 3, 3, 150000),

('2023-07-05', 4, 5, 4, 250000),

('2023-03-30', 5, 1, 5, 300000);

-- Sakatlik Tablosu --

CREATE TABLE Sakatlik (

sakatlikID INT IDENTITY(1,1) PRIMARY KEY,

oyuncuID INT,

yaralanmaTipi VARCHAR(100),

baslangicTarihi DATE,

bitisTarihi DATE,

FOREIGN KEY (oyuncuID) REFERENCES Oyuncu(oyuncuID)

);

INSERT INTO Sakatlik (oyuncuID, yaralanmaTipi, baslangicTarihi, bitisTarihi)

VALUES

(1, 'Diz Yaralanması', '2023-02-01', '2023-03-01'),

(2, 'Kas Yaralanması', '2023-04-10', '2023-05-10'),

(3, 'Ayak Bileği Burkulması', '2023-06-15', '2023-07-15'),

(4, 'Omuz Çıkığı', '2023-08-20', '2023-09-20'),

(5, 'Adale Çekmesi', '2023-10-01', '2023-10-31');

-- Rol Tablosu --

```
CREATE TABLE Rol (  
    rolID INT IDENTITY(1,1) PRIMARY KEY,  
    rolAdi VARCHAR(50)  
);
```

```
INSERT INTO Rol (rolAdi)  
VALUES  
( 'Teknik Direktör'),  
( 'Analist'),  
( 'Gözlemci'),  
( 'Futbolcu'),  
( 'Yönetici');
```

-- Scout Tablosu --

```
CREATE TABLE Scout (  
    scoutID INT IDENTITY(1,1) PRIMARY KEY,  
    ad VARCHAR(50),  
    soyad VARCHAR(50),  
    uzmanlikAlani VARCHAR(100),  
    calismaBolgesi VARCHAR(100),  
    rolID INT,  
    FOREIGN KEY (rolID) REFERENCES Rol(rolID)  
);
```

```
INSERT INTO Scout (ad, soyad, uzmanlikAlani, calismaBolgesi, rolID)  
VALUES  
( 'Can', 'Kara', 'Forvet', 'Türkiye', 3),
```

('James', 'Williams', 'Defans', 'İngiltere', 3),
('Marco', 'Bianchi', 'Kaleci', 'İtalya', 3),
('Miguel', 'Sanchez', 'Orta Saha', 'İspanya', 3),
('Karl', 'Schmidt', 'Kanat', 'Almanya', 3);

-- Rapor Tablosu --

```
CREATE TABLE Rapor (  
    raporID INT IDENTITY(1,1) PRIMARY KEY,  
    raporTarihi DATE,  
    icerik TEXT,  
    scoutID INT,  
    oyuncuID INT,  
    FOREIGN KEY (scoutID) REFERENCES Scout(scoutID),  
    FOREIGN KEY (oyuncuID) REFERENCES Oyuncu(oyuncuID)  
);
```

```
INSERT INTO Rapor (raporTarihi, icerik, scoutID, oyuncuID)  
VALUES  
( '2023-01-10', 'Hızlı ve güçlü bir forvet.', 1, 1),  
( '2023-02-15', 'Savunmada güven veriyor.', 2, 2),  
( '2023-03-20', 'Tekniği üst düzeyde.', 3, 3),  
( '2023-04-25', 'Top dağıtımı harika.', 4, 4),  
( '2023-05-30', 'Kanat bindirmeleri etkili.', 5, 5);
```

-- Stadyum Tablosu --

```
CREATE TABLE Stadyum (  
    StadyumID INT IDENTITY(1,1) PRIMARY KEY,  
    StadyumAdi VARCHAR(100),  
    Sehir VARCHAR(100),  
    Kapasite INT  
);
```

```
INSERT INTO Stadyum (StadyumAdi, Sehir, Kapasite)  
VALUES  
('Türk Telekom Arena', 'İstanbul', 50000),  
('Old Trafford', 'Manchester', 76000),  
('Allianz Stadium', 'Torino', 41000),  
('Santiago Bernabeu', 'Madrid', 81000),  
('Allianz Arena', 'Münih', 75000);
```

-- Mac Tablosu --

```
CREATE TABLE Mac (  
    MacID INT IDENTITY(1,1) PRIMARY KEY,  
    Tarih DATE,  
    StadyumID INT,  
    Takim1ID INT,  
    Takim2ID INT,  
    Sonuc VARCHAR(50),  
    FOREIGN KEY (StadyumID) REFERENCES Stadyum(StadyumID),  
    FOREIGN KEY (Takim1ID) REFERENCES Takim(TakimID),  
    FOREIGN KEY (Takim2ID) REFERENCES Takim(TakimID)  
);
```

INSERT INTO Mac (Tarih, StadyumID, Takim1ID, Takim2ID, Sonuc)

VALUES

('2023-01-15', 1, 1, 2, '2-1'),

('2023-02-20', 2, 3, 4, '1-1'),

('2023-03-25', 3, 5, 1, '0-3'),

('2023-04-30', 4, 2, 3, '1-0'),

('2023-05-10', 5, 4, 5, '3-2');

-- Performas Kaydi Tablosu --

CREATE TABLE PerformansKaydi (

PerformansID INT IDENTITY(1,1) PRIMARY KEY,

Tarih DATE,

OyuncuID INT,

MacID INT,

Dakika INT,

GolSayisi INT,

AsistSayisi INT,

PozisyonID INT,

ScoutYorum TEXT,

FOREIGN KEY (OyuncuID) REFERENCES Oyuncu(OyuncuID),

FOREIGN KEY (MacID) REFERENCES Mac(MacID),

FOREIGN KEY (PozisyonID) REFERENCES Pozisyon(PozisyonID)

);

INSERT INTO PerformansKaydi (Tarih, OyuncuID, MacID, Dakika, GolSayisi, AsistSayisi, PozisyonID, ScoutYorum)

VALUES

*('2023-01-15', 1, 1, 90, 2, 1, 4, 'Harika performans.'),
('2023-02-20', 2, 2, 85, 0, 0, 3, 'Savunmada başarılı.'),
('2023-03-25', 3, 3, 70, 1, 0, 5, 'Tekniği çok iyi.'),
('2023-04-30', 4, 4, 80, 0, 1, 2, 'Pas dağılımı etkili.'),
('2023-05-10', 5, 5, 75, 1, 2, 1, 'Hücumda aktif.');*

– MacIstatistikleri Tablosu –

*CREATE TABLE MacIstatistikleri (
IstatistikID INT IDENTITY(1,1) PRIMARY KEY,
MacID INT,
SutSayisi INT,
FaulSayisi INT,
KirmiziKart INT,
SariKart INT,
FOREIGN KEY (MacID) REFERENCES Mac(MacID)
);*

*INSERT INTO MacIstatistikleri (MacID, SutSayisi, FaulSayisi, KirmiziKart, SariKart)
VALUES
(1, 10, 15, 0, 2),
(2, 12, 18, 1, 3),
(3, 8, 10, 0, 1),
(4, 14, 12, 2, 4),
(5, 9, 20, 0, 5);*

GO;

-- PROCEDURE Örneği --

Bir oyuncu transferi sırasında eski ve yeni takımın kadro değerini güncelleyen bir örnek :

CREATE PROCEDURE TransferOyuncu

@OyuncuID INT,

@YeniTakimID INT,

@TransferUcreti DECIMAL(15, 2),

@TransferTarihi DATE

AS

BEGIN

BEGIN TRANSACTION;

DECLARE @EskiTakimID INT;

SELECT @EskiTakimID = takimID FROM Oyuncu WHERE oyuncuID = @OyuncuID;

INSERT INTO TransferKaydi (transferTarihi, eskiTakimID, yeniTakimID, oyuncuID, transferUcreti)

VALUES (@TransferTarihi, @EskiTakimID, @YeniTakimID, @OyuncuID, @TransferUcreti);

UPDATE Oyuncu

SET takimID = @YeniTakimID

WHERE oyuncuID = @OyuncuID;

UPDATE Takim

SET deger = deger - @TransferUcreti

WHERE takimID = @EskiTakimID;

UPDATE Takim

SET deger = deger + @TransferUcreti

WHERE takimID = @YeniTakimID;

COMMIT TRANSACTION;

END;

Örnek kullanımımı:

*EXEC TransferOyuncu @OyunculD = 5, @YeniTakimID = 1, @TransferUcreti = 500000,
@TransferTarihi = '2025-01-01';*

-- TRIGGER Örneği --

*Bu işlem sonrasında, TransferKaydi tablosuna veri eklendiğinde
TransferSonrasiTakimDegerGuncelle trigger'ı otomatik çalışacak ve takım
değerlerini güncelleyecektir:*

CREATE TRIGGER TransferSonrasiTakimDegerGuncelle

ON TransferKaydi

AFTER INSERT

AS

BEGIN

DECLARE @EskiTakimID INT, @YeniTakimID INT, @TransferUcreti DECIMAL(15, 2);

SELECT @EskiTakimID = eskiTakimID,

@YeniTakimID = yeniTakimID,

@TransferUcreti = transferUcreti

FROM INSERTED;

UPDATE Takim

SET deger = deger - @TransferUcreti

WHERE takımID = @EskiTakimID;

UPDATE Takim

SET deger = deger + @TransferUcreti

WHERE takımID = @YeniTakimID;

END;

Örnek kullanımımı:

INSERT INTO TransferKaydi (transferTarihi, eskiTakimID, yeniTakimID, oyuncuID, transferUcreti)

VALUES ('2025-01-04', 1, 5, 5, 500000);

-- TRANSACTION Örneği--

Bu işlem sonrasında eğer transfer kaydı başarılı bir şekilde gerçekleşir ise ekrana 'Transfer işlemi başarıyla tamamlandı.' şeklinde bir çıktı çıkar.

Fakat herhangi bir hata ile karşılaşıldığında hataya kadar yapılan bütün değişiklikler geri alınır ve ekranda 'Bir hata oluştu. Tüm değişiklikler geri alındı.' mesajını görürüz.

BEGIN TRANSACTION;

BEGIN TRY

DECLARE @EskiTakimID INT, @YeniTakimID INT, @TransferUcreti DECIMAL(15, 2), @OyuncuID INT;

SET @OyuncuID = 2; -- Transfer edilecek oyuncunun ID'si

SET @YeniTakimID = 1; -- Oyuncunun gideceği yeni takım

SET @TransferUcreti = 500000; -- Transfer ücreti

SELECT @EskiTakimID = takımID FROM Oyuncu WHERE oyuncuID = @OyuncuID;

INSERT INTO TransferKaydi (transferTarihi, eskiTakimID, yeniTakimID, oyuncuID, transferUcreti)

VALUES (GETDATE(), @EskiTakimID, @YeniTakimID, @OyuncuID, @TransferUcreti);

UPDATE Oyuncu

SET takimID = @YeniTakimID

WHERE oyuncuID = @OyuncuID;

UPDATE Takim

SET deger = deger - @TransferUcreti

WHERE takimID = @EskiTakimID;

UPDATE Takim

SET deger = deger + @TransferUcreti

WHERE takimID = @YeniTakimID;

COMMIT TRANSACTION;

PRINT 'Transfer işlemi başarıyla tamamlandı.';

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION;

PRINT 'Bir hata oluştu. Tüm değişiklikler geri alındı.';

PRINT ERROR_MESSAGE();

END CATCH;

10.2. Örnek sorgu kodları:

--Bir oyuncunun hangi takımda oynadığını, hangi pozisyonda olduğunu ve menajer bilgilerini listeleyin.

```
SELECT
    O.ad AS OyuncuAdi,
    O.soyad AS OyuncuSoyadi,
    T.takimAdi AS TakimAdi,
    P.pozisyonAdi AS Pozisyon,
    M.ad AS MenajerAdi,
    M.soyad AS MenajerSoyadi,
    M.telefon AS MenajerTelefon
FROM
    Oyuncu O
JOIN
    Takim T ON O.takimID = T.takimID
JOIN
    OyuncuPozisyon OP ON O.oyuncuID = OP.oyuncuID
JOIN
    Pozisyon P ON OP.pozisyonID = P.pozisyonID
JOIN
    Menajer M ON T.menajerID = M.menajerID;

GO
```

-- Belirli bir oyuncunun ("Mehmet Kaya") transfer kayıtlarını listeleyin.

```
SELECT
    O.ad AS OyuncuAdi,
    O.soyad AS OyuncuSoyadi,
    ET.takimAdi AS EskiTakim,
    YT.takimAdi AS YeniTakim,
    T.transferTarihi,
    T.transferUcreti
FROM
    TransferKaydi T
JOIN
    Oyuncu O ON T.oyuncuID = O.oyuncuID
JOIN
    Takim ET ON T.eskiTakimID = ET.takimID
JOIN
    Takim YT ON T.yeniTakimID = YT.takimID
WHERE
    O.ad = 'Mehmet' AND O.soyad = 'Kaya';

GO
```

-- Bir scout tarafından hazırlanan oyuncu raporlarını listeleyin.

```
SELECT
    S.ad AS ScoutAdi,
    S.soyad AS ScoutSoyadi,
    S.uzmanlikAlani,
    R.raporTarihi,
    R.icerik AS Rapor,
    O.ad AS OyuncuAdi,
    O.soyad AS OyuncuSoyadi
FROM
    Rapor R
JOIN
    Scout S ON R.scoutID = S.scoutID
JOIN
    Oyuncu O ON R.oyuncuID = O.oyuncuID
```

```

WHERE
    S.ad = 'Can' AND S.soyad = 'Kara';

GO

-- Bir takımın toplam piyasa değeri ve en değerli oyuncusu.

SELECT
    T.takimAdi AS TakimAdi,
    SUM(O.deger) AS ToplamOyuncuDegeri,
    MAX(O.deger) AS EnDegerliOyuncuDegeri,
    MAX(O.ad) + ' ' + MAX(O.soyad) AS EnDegerliOyuncu
FROM
    Oyuncu O
JOIN
    Takim T ON O.takimID = T.takimID
GROUP BY
    T.takimAdi;

GO

-- Oyuncuların attıkları gol sayısı ve oynadıkları takımlar.
SELECT
    O.ad AS OyuncuAdi,
    O.soyad AS OyuncuSoyadi,
    T.takimAdi AS TakimAdi,
    SUM(PK.GolSayisi) AS ToplamGol
FROM
    PerformansKaydi PK
JOIN
    Oyuncu O ON PK.OyuncuID = O.oyuncuID
JOIN
    Takim T ON O.takimID = T.takimID
GROUP BY
    O.ad, O.soyad, T.takimAdi
ORDER BY
    ToplamGol DESC;

GO

-- Bir maçın detayları: Takımlar, skor, stadyum ve tarih.(MacId ile sorgulama)
SELECT
    M.Tarih,
    ST.StadyumAdi,
    ST.Sehir,
    T1.takimAdi AS Takim1,
    T2.takimAdi AS Takim2,
    M.Sonuc
FROM
    Mac M
JOIN
    Stadyum ST ON M.StadyumID = ST.StadyumID
JOIN
    Takim T1 ON M.Takim1ID = T1.takimID
JOIN
    Takim T2 ON M.Takim2ID = T2.takimID
WHERE
    M.MacID = 1;

GO

```

```
-- Bir ligdeki takımlar ve toplam oyuncu sayıları.(LigAdi ile sorgulama)
SELECT
    L.ligAdi AS LigAdi,
    T.takimAdi AS TakimAdi,
    COUNT(O.oyuncuID) AS OyuncuSayisi
FROM
    Lig L
JOIN
    Takim T ON L.ligID = T.ligID
JOIN
    Oyuncu O ON T.takimID = O.takimID
WHERE
    L.ligAdi = 'Süper Lig'
GROUP BY
    L.ligAdi, T.takimAdi;
```