yazı tipi, metin, grafik, tipografi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**

**KÜTÜPHANE YÖNETİM SİSTEMİ**

**Veritabanı Sistemleri**

**Doç. Dr. Selçuk Kıran**

insan yüzü, kişi, şahıs, adam, insan, giyim içeren bir resim

Açıklama otomatik olarak oluşturuldu

Beyzanur Baki                                                 Kerem Yılmaz

2104040019                                                                   2204040176

insan yüzü, kişi, şahıs, gülümsemek, gülüş, portre içeren bir resim

Açıklama otomatik olarak oluşturuldukişi, şahıs, insan yüzü, gülümsemek, gülüş, giyim içeren bir resim

Açıklama otomatik olarak oluşturuldu

Özlem Babatutmaz                               Melike Aydoğduoğlu

2204040149                                                  2204040137

Şerefim ve inandığım tüm değerler adına yemin ederim ki bu sınavda hiçbir yerden kopya çekmedim ve etik kurallarına karşı gelmedim.

1. **Kapsam Tanımı**

Kütüphane yönetim sistemi, kitap envanterinin dijital olarak takibi, kullanıcı hesaplarının yönetimi, ödünç alma ve iade süreçlerinin izlenmesi, cezai işlemlerin kaydedilmesi gibi temel işlemleri kapsayacaktır.

Sistem sayesinde, kütüphane üyeleri online olarak kitapların durumunu (mevcut/ödünç alınmış) kontrol edebilecek ve geçmiş işlem bilgilerine ulaşabilecektir. Aynı zamanda kütüphane personeli kitap envanterini güncelleyebilecek, kullanıcı hesaplarını yönetebilecek ve sistem raporlarını alabilecektir.

**1.1. Sistem Gereksinimlerinin Belirlenmesi**

* Kullanıcılar sisteme kayıt olabilir ve giriş yapabilir.
* Her bir kitaptan yalnızca bir adet bulunmaktadır.
* Kitap bilgileri (ISBN, başlık, yazar, tür, yayın yılı vb.) sisteme kaydedilecektir.
* Kitapların mevcut durumu (ödünçte/mevcut) kullanıcılar tarafından görülebilecektir.
* Kitap ödünç alma ve iade işlemleri sistemde kayıt altına alınacaktır.
* Kullanıcı işlem geçmişi ve kitap envanter durumu hakkında raporlar alınabilecektir.
* Personel, kitap ekleyebilir/silebilir ve envanteri düzenleyebilir.

**1.2. Ana Kısıtlar**

**1.Veri Tutarlılığı:**

* Aynı kitabın birden fazla kullanıcıya aynı anda ödünç verilmesi engellenmelidir. Çünkü her kitaptan yalnızca bir adet bulunmaktadır.

**2.Uyumluluk:**

* Sistem hem web tarayıcılarında hem de mobil cihazlarda kullanılabilir olmalıdır.

**3.Ceza** **Sistemi**:

* Kullanıcıya ait ceza varsa kullanıcı bir kitap daha ödünç alamaz.
* Geciken iadeler için cezai işlemler 30 gün sonrasında başlayacaktır. Gecikme sonrası kullanıcının kitabı vermediği her geçen gün 5 TL artacak şekilde işlenecektir.

**4.Veritabanı Kısıtları:**

* Her bir kitap yalnızca bir türe ait olabilir.
* Her kitabın sadece bir yazarı ait olabilir.
* Kullanıcının iade edilmeyen bir kitabı varsa başka bir kitap alamaz.

**1.3. Kavramsal Tasarım**

Sistem aşağıdaki temel varlıkları içerecektir:

**1. Üyeler (Members):**

- Ad (MFIRST\_NAME), soyad (MLAST\_NAME), e-posta (EMAIL), telefon (MPHONE\_NUMBER), şehir (CITY\_ID), üyelik tarihi (MEMBERSHIP\_DATE) bilgilerini içerecektir.

- Her üyeye sistem tarafından benzersiz bir kimlik numarası (MEMBER\_ID) atanacaktır.

**2. Kitaplar (Books):**

- Kitap adı (TITLE), ISBN numarası (ISBN), yazar (AUTHOR\_ID), tür (GENRE\_ID), yayın yılı (PUBLICATION\_YEAR), yayınevi (PUBLISHER\_ID), durum (BOOK\_STATUS) bilgilerini içerecektir.

- Her kitap için benzersiz bir kimlik numarası (BOOK\_ID) olacaktır.

**3. Yazarlar (Authors):**

- Ad (AFIRST\_NAME), soyad (ALAST\_NAME), doğum tarihi (ABIRTH\_DATE) bilgileri kaydedilecektir.

- Her yazar için benzersiz bir kimlik numarası (AUTHOR\_ID) olacaktır.

**4. Türler (Genres):**

- Tür adı (GENRE\_NAME) bilgisini içerecektir.

- Her türe sistem tarafından benzersiz bir kimlik numarası (GENRE\_ID) atanacaktır.

**5. İşlem Geçmişi (Borrowings):**

- Kullanıcıların ödünç aldığı kitapların işlem bilgileri kaydedilecektir.

- Ödünç alınan kitap (BOOK\_ID), kullanıcı (MEMBER\_ID), ödünç alma tarihi (BORROW\_DATE), iade tarihi (RETURN\_DATE), iade edilmesi gereken tarih (DUE\_DATE), ceza miktarı (PENALTY\_AMOUNT), cezayı kesen çalışan ID (STAFF\_ID) gibi bilgiler bulunacaktır.

- Her işlem için benzersiz bir kimlik numarası (BORROWING\_ID) olacaktır.

**6. Personel (Staff):**

- Ad (SFIRST\_NAME), soyad (SLAST\_NAME), telefon (SPHONE\_NUMBER), pozisyon bilgisi (POSITION\_ID) gibi bilgiler içerecektir.

- Her personele sistem tarafından benzersiz bir kimlik numarası (STAFF\_ID) atanacaktır.

**7. Cezalar (Penalties):**

- Gecikmiş iadeler veya kaybolmuş kitaplar için ceza bilgileri kaydedilecektir.

- Kullanıcı (MEMBER\_ID), kitap (BOOK\_ID), iade tarihi (RETURN\_DATE), ceza miktarı (PENALTY\_AMOUNT), iade edilmesi gereken tarih (DUE\_DATE) ,  durum (PENALTY\_STATUS) bilgileri yer alacaktır.

-Her ceza bilgisine sistem tarafından benzersiz bir ceza kimlik numarası (PENALTY\_ID) atanacaktır.

**8. Şehirler (Cities):**

- Şehir adı (CITY\_NAME) bilgisi tutulacaktır.

- Her şehir için benzersiz bir kimlik numarası (CITY\_ID) olacaktır.

**9. Yayınevleri (Publishers):**

- Yayınevi adı (PUBLISHER\_NAME), telefon numarası (PHONE\_NUMBER) bilgileri yer alacaktır.

- Her yayınevine sistem tarafından benzersiz bir kimlik numarası (PUBLISHER\_ID) atanacaktır.

**10. Pozisyonlar (Positions):**

- Personel pozisyon bilgisi (POSITION\_NAME) yer alacaktır.

- Her pozisyon için benzersiz bir kimlik numarası (POSITION\_ID) olacaktır.

**1.4. Potansiyel Sistem Kullanıcıları**

* **Kütüphane Personeli:** Envanter yönetimi ve sistem raporlaması gerçekleştirir.
* **Yönetim:** Genel raporlama ve sistem denetimini yapar.

**2. Data Dictionary**

metin, makbuz, sayı, numara, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, yazı tipi, ekran görüntüsü, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, makbuz, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**3-Entity Relationship Model Diagram**

metin, diyagram, plan, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu

**4- SQL Sorguları**

* 32 SQL Sorgusu

**1-Üyelerin aldıkları kitap sayısını ve toplam gecikme cezalarını hesaplayıp bu üyelerden toplam ceza miktarı 5 TL den fazla olanlar**

SELECT

M.Member\_ID, M.MFirst\_Name + ' ' + M.MLast\_Name AS Member\_Name,

  COUNT(B.Book\_ID) AS Total\_Borrowed\_Books,

   SUM(P.Penalty\_Amount) AS Total\_Penalty

FROM Members M

LEFT JOIN Borrowings B ON M.Member\_ID = B.Member\_ID

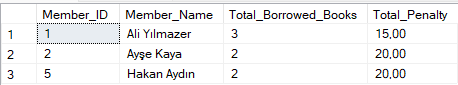
LEFT JOIN Penalties P ON M.Member\_ID = P.Member\_ID

GROUP BY

M.Member\_ID, M.MFirst\_Name, M.MLast\_Name

HAVING

            SUM(P.Penalty\_Amount) > 5;



**2-Her yayıncının yayınladığı kitap sayısını gösteren bir sorgu:**

SELECT

     P.Publisher\_Name,

    COUNT(B.Book\_ID) AS Total\_Books

FROM

Publishers P

INNER JOIN Books B ON P.Publisher\_ID = B.Publisher\_ID

GROUP BY

   P.Publisher\_Name

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**3-Hangi ilden toplam ne kadar üye olduğunu gösteren sorgu:**

SELECT C.City\_Name,  COUNT(M.Member\_ID) AS Total\_Members

FROM  Cities C

LEFT JOIN Members M ON C.City\_ID = M.City\_ID

GROUP BY  C.City\_Name

ORDER BY

Total\_Members DESC;

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**4- Her türdeki kitapların mevcut durumu ve toplam sayısı:**

SELECT

G.Genre\_Name,

SUM(CASE WHEN B.Book\_Status = 1 THEN 1 ELSE 0 END) AS Available\_Books,

SUM(CASE WHEN B.Book\_Status = 0 THEN 1 ELSE 0 END) AS Borrowed\_Books,

COUNT(B.Book\_ID) AS Total\_Books

FROM

Genres G

LEFT JOIN Books B ON G.Genre\_ID = B.Genre\_ID

GROUP BY

G.Genre\_Name

HAVING

COUNT(B.Book\_ID) > 1;

**metin, yazı tipi, ekran görüntüsü, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**5- Her personelin kontrol ettiği kitap sayısını listeleyen bir sorgu:**

SELECT

S.Staff\_ID, S.SFirst\_Name + ' ' + S.SLast\_Name AS Staff\_Name,

COUNT(DISTINCT B.Book\_ID) AS Processed\_Books

FROM Staff S

INNER JOIN Borrowings B ON S.Staff\_ID = B.Staff\_ID

GROUP BY

S.Staff\_ID, S.SFirst\_Name, S.SLast\_Name

HAVING

COUNT(DISTINCT B.Book\_ID) > 2;



**6-Borcu olan üyelerin hangi illerde oturduğu**

SELECT

C.City\_Name,

COUNT(DISTINCT M.Member\_ID) AS Total\_Borrowing\_Members,

SUM(P.Penalty\_Amount) AS Total\_Penalty\_Amount

FROM Cities C

INNER JOIN Members M ON C.City\_ID = M.City\_ID

INNER JOIN Penalties P ON M.Member\_ID = P.Member\_ID

WHERE

P.Penalty\_Amount > 0

GROUP BY

C.City\_Name

metin, yazı tipi, ekran görüntüsü, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

**7- Kütüphanemizdeki yazarların kitap sayısının artacak şekilde listelenmesi**

SELECT

A.AFirst\_Name +' ' +A.ALast\_Name AS Author\_Name,

COUNT(B.Book\_ID) AS Total\_Books\_Written

FROM Authors A

INNER JOIN Books B ON A.Author\_ID = B.Author\_ID

GROUP BY

A.AFirst\_Name, A.ALast\_Name

ORDER BY

Total\_Books\_Written ASC;

**metin, yazı tipi, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**8-Kitabın adını türünü ve bu kitaplara ait ne kadar ceza olduğunu gösteren sorgu**

SELECT

B.Title AS Book\_Title, G.Genre\_Name AS Genre, Bor.Penalty\_Amount AS Penalty\_Amount

FROM Borrowings Bor

INNER JOIN Books B ON Bor.Book\_ID = B.Book\_ID

INNER JOIN Genres G ON B.Genre\_ID = G.Genre\_ID

WHERE

Bor.Penalty\_Amount > 0

ORDER BY

G.Genre\_Name, B.Title;



**9-Kitapların yazar isim soyisimleriyle listelenmesi**

SELECT b.Title AS Book\_Title,

 a.AFirst\_Name AS Author\_First\_Name, a.ALast\_Name AS Author\_Last\_Name

FROM  Books b

JOIN Authors a ON b.Author\_ID = a.Author\_ID;



**10-Kitap Türlerini ve Yayıncı İsimlerini Listeleme**

SELECT g.Genre\_Name AS Genre,

p.Publisher\_Name AS Publisher

FROM Books b

JOIN

Genres g ON b.Genre\_ID = g.Genre\_ID

JOIN

Publishers p ON b.Publisher\_ID = p.Publisher\_ID;



**11- Her yazarın kaç kitabı olduğunu öğrenmek için Books ve Authors tablolarını gruplandırarak yazar bazında kitap sayısını listeler.**

SELECT A.AFirst\_Name ,A.ALast\_Name,

COUNT(B.Book\_ID) AS Book\_Count

FROM Books B

JOIN Authors A ON B.Author\_ID = A.Author\_ID

GROUP BY

A.AFirst\_Name, A.ALast\_Name;



**12-EXISTS kullanarak cezalı üyeleri listeleme**

SELECT MFirst\_Name, MLast\_Name

FROM Members m

WHERE

EXISTS (SELECT 1

FROM Penalties p

WHERE p.Member\_ID = m.Member\_ID

AND p.Penalty\_Status = 1

);

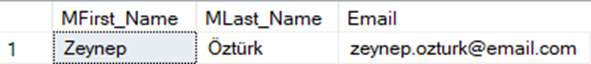


**13-IN:Belirli Bir Şehirdeki Üyelerin Listesi**

SELECT MFirst\_Name, MLast\_Name, Email

FROM Members

WHERE City\_ID IN ( SELECT City\_ID FROM CitiesWHERE City\_Name = 'Ankara');



**14-UNION: Farklı Şehirlerdeki Personel ve Üyeleri Listeleme**

SELECT MFirst\_Name AS First\_Name, MLast\_Name AS Last\_Name, 'Member' AS Role

FROM Members

WHERECity\_ID = 1

UNION (SELECT SFirst\_Name AS First\_Name, SLast\_Name AS Last\_Name,'Staff' AS Role

FROM Staff

WHERE Position\_ID = 2);

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**15-INTERSECT: Ceza Almış Üyelerin Listesi**

SELECT Member\_ID FROM Borrowings

WHERE Penalty\_Amount > 0

INTERSECT

SELECT Member\_ID

FROM Penalties

WHERE Penalty\_Status = 1;

ekran görüntüsü, metin, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

**16-Üyeler ve şehir bilgileri**

SELECT

m.MFirst\_Name + ' ' + m.MLast\_Name AS Uye\_Adi,

c.City\_Name AS Sehir

FROM Members m

JOIN Cities c ON m.City\_ID = c.City\_ID;



**17-Personellerin ve Pozisyonlarının Listelenmesi**

SELECT

s.SFirst\_Name + ' ' + s.SLast\_Name AS Personel\_Adi,

p.Position\_Name AS Pozisyon

FROM Staff s

JOIN Positions p ON s.Position\_ID = p.Position\_ID;



18-**Yazarlar ve Doğum Tarihleri**

SELECT

a.AFirst\_Name + ' ' + a.ALast\_Name AS Yazar\_Adi,

a.ABirth\_Date AS Dogum\_Tarihi

FROM Authors a;



19-**Üyeler ve Personel Telefon Numaraları**

SELECT

m.MPhone\_Number AS Telefon,

'Uye' AS Tip

FROM Members m

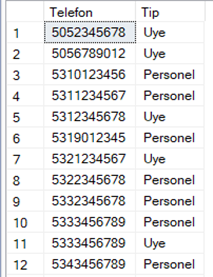
UNION

SELECT

s.SPhone\_Number AS Telefon,

'Personel' AS Tip

FROM Staff s;



**20-Üye ve Personel Adları**

SELECT

m.MFirst\_Name + ' ' + m.MLast\_Name AS Ad

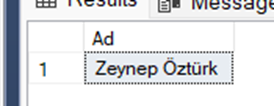
FROM Members m

INTERSECT

SELECT

s.SFirst\_Name + ' ' + s.SLast\_Name AS Ad

FROM Staff s;



**21-Pozisyonlar ve Türler Birleşimi**

SELECT

p.Position\_Name AS Isim,

'Pozisyon' AS Tip

FROM Positions p

UNION

SELECT

g.Genre\_Name AS Isim,

'Kitap Turu' AS Tip

FROM Genres g;



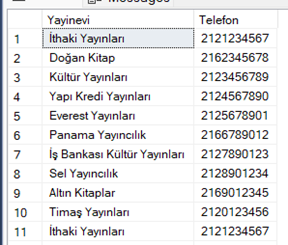
22-**Yayınevleri ve İsimleri**

SELECT

    p.Publisher\_Name AS Yayinevi,

p.PPhone\_Number AS Telefon

FROM Publishers p;



23- **Üyelerin Şehirleri ve Onları Kaydeden Şehirlerin Personeli**

SELECT

m.MFirst\_Name + ' ' + m.MLast\_Name AS Uye\_Adi,

c.City\_Name AS Sehir,

s.SFirst\_Name + ' ' + s.SLast\_Name AS Personel\_Adi

FROM Members m

JOIN Cities c ON m.City\_ID = c.City\_ID

JOIN Staff s ON s.Staff\_ID = c.City\_ID; -- Şehir ve personel arasında hayali ilişki



**24-Personel ve Pozisyonlarının Sayısı**

SELECT

p.Position\_Name AS Pozisyon,

COUNT(s.Staff\_ID) AS Personel\_Sayisi

FROM Positions p

LEFT JOIN Staff s ON p.Position\_ID = s.Position\_ID

GROUP BY p.Position\_Name;

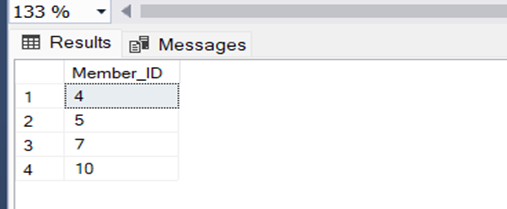


**25. Kitap ödünç alan ancak ceza almayan üyeler**

SELECT Member\_ID FROM Borrowings

EXCEPT

SELECT Member\_ID FROM Penalties;



**26.Ödünç alınan ancak iade edilmeyen kitaplar**

SELECT Book\_ID FROM Borrowings WHERE Return\_Date IS NULL

EXCEPT

SELECT Book\_ID FROM Borrowings WHERE Return\_Date IS NOT NULL;

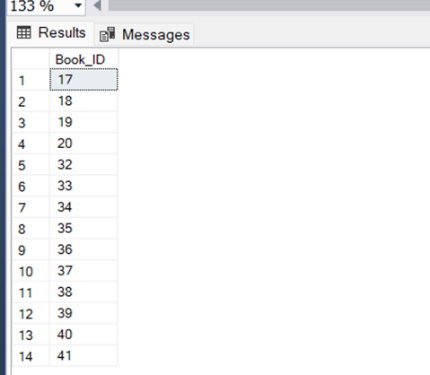


**27. Bir türe ait olan ancak ödünç alınmayan kitaplar**

SELECT Book\_ID FROM Books

EXCEPT

SELECT Book\_ID FROM Borrowings;

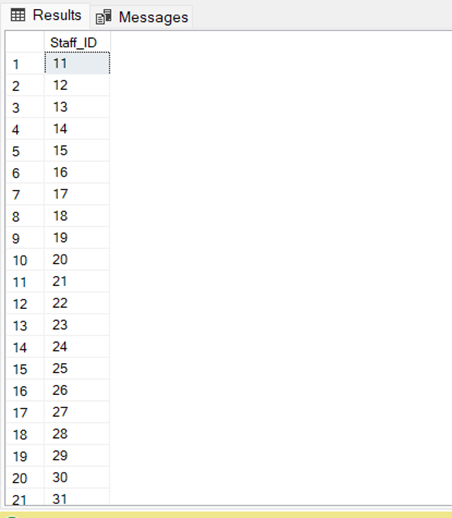


**28. Göreve atanan, ancak ödünç kayıtları olmayan personel üyeleri**

SELECT Staff\_ID FROM Staff

EXCEPT

SELECT Staff\_ID FROM Borrowings;



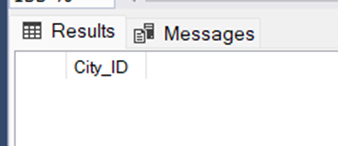
**29. Üyeleri olan ancak borçlanmayan şehirler**

SELECT City\_ID FROM Members

EXCEPT

SELECT City\_ID FROM Borrowings B

INNER JOIN Members M ON B.Member\_ID = M.Member\_ID;



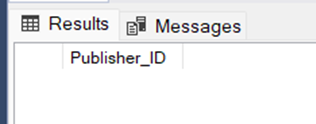
**30. Asla ödünç alınmayan kitaplara sahip yayıncılar**

SELECT Publisher\_ID FROM Books

EXCEPT

SELECT Publisher\_ID FROM Borrowings B

INNER JOIN Books Bo ON B.Book\_ID = Bo.Book\_ID;



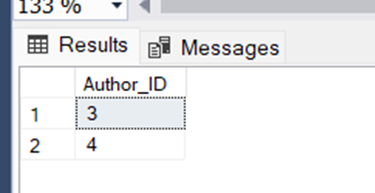
**31. Kütüphanede kitapları bulunan ancak ödünç alınmayan yazarlar**

SELECT Author\_ID FROM Books

EXCEPT

SELECT Author\_ID FROM Borrowings B

INNER JOIN Books Bo ON B.Book\_ID = Bo.Book\_ID;

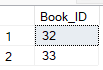


**32. Ödünç alınmış ancak iade edilmemiş kitaplar**

SELECT Book\_ID FROM Borrowings WHERE Return\_Date IS NULL

EXCEPT

SELECT Book\_ID FROM Borrowings WHERE Return\_Date IS NOT NULL;



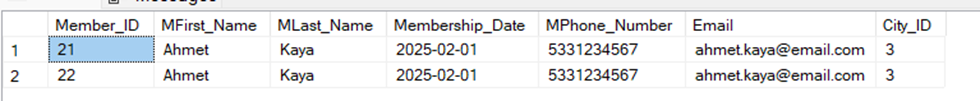
* **5 INSERT**

**1.**    **Yeni bir üye eklemek**

INSERT INTO Members (MFirst\_Name, MLast\_Name, Membership\_Date,

MPhone\_Number, Email, City\_ID)

VALUES ('Ahmet', 'Kaya', '2025-02-01', '5331234567', 'ahmet.kaya@email.com', 3);

SELECT \* FROM Members WHERE MFirst\_Name = 'Ahmet' AND MLast\_Name = 'Kaya';

**2.    Yeni bir kitap eklemek**

INSERT INTO Books (Title, ISBN, Publicition\_Year, Publisher\_ID, Book\_Status, Genre\_ID, Author\_ID)

VALUES ('Yeni Dünya', '9789750800000', '2005-01-01', 2, 1, 3, 5);

select \* from books where Title= 'Yeni Dünya' ;



3.**Yeni bir personel eklemek:**

INSERT INTO Staff (SFirst\_Name, SLast\_Name, Position\_ID, SPhone\_Number)

VALUES ('Fatma', 'Yılmaz', 4, '5321236789');

select \* from staff where Position\_ID = 4 ;



4. **Yeni bir şehir eklemek:**

INSERT INTO Cities (City\_ID, City\_Name)

VALUES (9, 'Trabzon');

SELECT City\_Name FROM Cities WHERE City\_Name LIKE 'T%';



5- **Yeni bir yazar eklemek:**

INSERT INTO Authors (AFirst\_Name, ALast\_Name, ABirth\_Date)

VALUES ('Elif', 'Şafak', '1971-10-25');

SELECT ABirth\_Date,AFirst\_Name from Authors WHERE ABirth\_Date LIKE '1971-10-25';



**6. Yeni bir ödünç alma eklemek:**

INSERT INTO Borrowings (Book\_ID, Member\_ID, Borrow\_Date, Due\_Date, Return\_Date, Staff\_ID, Penalty\_Amount)

VALUES

(32, 1, '2024-10-21', '2024-11-27', NULL, 11, 0), -- Kitap henüz iade edilmedi

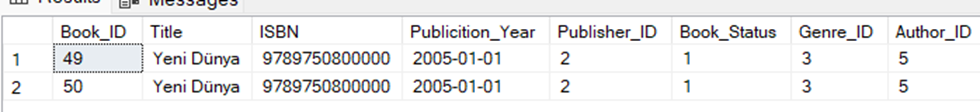
(33, 2, '2024-10-22', '2024-11-28', NULL, 12, 0); -- Kitap henüz iade edilmedi



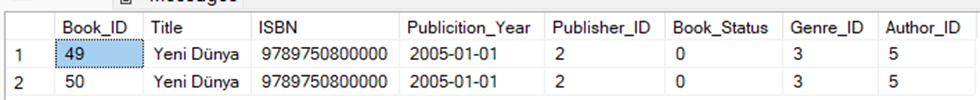
* **5 UPDATE**

1-**Bir kitabın durumunu (ödünç alınmış mı?) güncellemek:**

Select \* from Books where Title = 'Yeni Dünya';



UPDATE Books SET Book\_Status = 0 -- Ödünç alınmış WHERE Title = 'Yeni Dünya';



**2-Bir personelin pozisyonunu güncellemek:**

Select \* from staff WHERE SFirst\_Name = 'Fatma' AND SLast\_Name = 'Yılmaz';

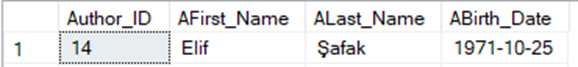


UPDATE Staff SET Position\_ID = 3 WHERE SFirst\_Name = 'Fatma' AND SLast\_Name = 'Yılmaz';



**3-Bir yazarın soyadını güncellemek:**

select \* from authors where AFirst\_Name = 'Elif';



UPDATE Authors SET ALast\_Name = 'Şafak-Gül' WHERE AFirst\_Name = 'Elif' AND ALast\_Name = 'Şafak';



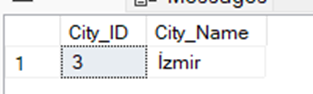
4- **Bir şehrin adını düzeltmek:**

select \* from Cities where  City\_ID = 3;

metin, yazı tipi, çizgi, sayı, numara içeren bir resim

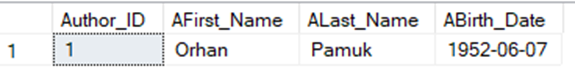
Açıklama otomatik olarak oluşturuldu

UPDATE Cities SET City\_Name = 'İzmir'  WHERE City\_ID = 3;



5-**Bir Yazarın Doğum Tarihini Güncelleme**

select \* from Authors where AFirst\_Name = 'Orhan' AND ALast\_Name = 'Pamuk';



UPDATE Authors SET ABirth\_Date = '1970-01-01' WHERE AFirst\_Name = 'Orhan' AND ALast\_Name = 'Pamuk';

metin, yazı tipi, çizgi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

* **2 DELETE**

**1-Bir üyeyi silmek**

SELECT \* from members WHERE MFirst\_Name = 'Ahmet' AND MLast\_Name = 'Kaya';



 DELETE FROM Members WHERE MFirst\_Name = 'Ahmet' AND MLast\_Name = 'Kaya';



**2-Kitabı silmek**

select \* from books WHERE Title = 'Yeni Dünya';



 DELETE FROM Books WHERE Title = 'Yeni Dünya';



* **2 TRIGGER**

**1-Bir kitap ödünç alındığında , ilgili kitabın Books tablosundaki durumunu "ödünç alınmış" olarak günceller.**

CREATE TRIGGER trg\_UpdateBookStatus

ON Borrowings

AFTER INSERT

AS

BEGIN

    UPDATE Books

    SET Book\_Status = 0 -- Ödünç alınmış

    WHERE Book\_ID IN (SELECT Book\_ID FROM INSERTED);

END;

**2-Bir kitap Books tablosundan silindiğinde, o kitaba ait tüm ceza kayıtlarını otomatik olarak siler.**

 CREATE TRIGGER trg\_DeleteBookPenalties

ON Books

AFTER DELETE

AS

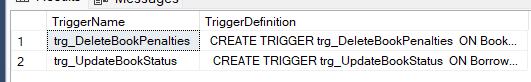
BEGIN

    DELETE FROM Penalties

    WHERE Book\_ID IN (SELECT Book\_ID FROM DELETED);

END;

**Her iki triggerın varlığının sorgusu;**



* **CREATE, ALTER ve DROP komutları**

CREATE DATABASE Kutuphane\_Yonetim\_Sistemi

CREATE TABLE Publishers(

Publisher\_ID INT IDENTITY(1,1) PRIMARY KEY,

Publisher\_Name VARCHAR(30),

PPhone\_Number CHAR(10)

)

CREATE TABLE Authors(

Author\_ID INT IDENTITY(1,1) Primary Key,

AFirst\_Name NVARCHAR(50),

ALast\_Name NVARCHAR(50),

ABirth\_Date DATE

)

CREATE TABLE Genres(

Genre\_ID INT IDENTITY(1,1) Primary Key,

Genre\_Name VARCHAR(50)

)

CREATE TABLE Cities(

City\_ID TINYINT PRIMARY KEY,

City\_Name VARCHAR(30)

)

CREATE TABLE Positions(

Position\_ID TINYINT IDENTITY(1,1) Primary Key,

Position\_Name VARCHAR(30)

)

CREATE TABLE Staff(

Staff\_ID TINYINT IDENTITY(1,1) Primary Key,

SFirst\_Name VARCHAR(30),

SLast\_Name VARCHAR(30),

Position\_ID TINYINT,

SPhone\_Number CHAR(10)

CONSTRAINT fk\_Position FOREIGN KEY (Position\_ID) REFERENCES Positions(Position\_ID)

)

CREATE TABLE Books (

Book\_ID INT IDENTITY(1,1) Primary Key,

Title VARCHAR(50),

ISBN CHAR(13),

Publicition\_Year DATE,

Publisher\_ID INT ,

Book\_Status BIT,

Genre\_ID INT,

Author\_ID INT,

CONSTRAINT fk\_Publisher FOREIGN KEY (Publisher\_ID) REFERENCES Publishers(Publisher\_ID),

CONSTRAINT fk\_Genre FOREIGN KEY (Genre\_ID) REFERENCES Genres(Genre\_ID),

CONSTRAINT fk\_Author FOREIGN KEY (Author\_ID) REFERENCES Authors(Author\_ID)

)

CREATE TABLE Members(

Member\_ID INT IDENTITY(1,1) Primary Key,

MFirst\_Name NVARCHAR(50),

MLast\_Name NVARCHAR(50),

Membership\_Date DATE,

MPhone\_Number CHAR(10),

Email VARCHAR(50),

City\_ID TINYINT

CONSTRAINT fk\_City FOREIGN KEY (City\_ID) REFERENCES Cities(City\_ID)

)

CREATE TABLE Borrowings(

    Borrowing\_ID INT IDENTITY(1,1) PRIMARY KEY,

    Book\_ID INT,

    Member\_ID INT,

    Borrow\_Date DATETIME2, --YYYY-MM-DD hh:mm:ss

    Due\_Date DATETIME2,

    Return\_Date DATETIME2,

    Penalty\_Amount MONEY,

    Staff\_ID TINYINT,

    CONSTRAINT fk\_Book FOREIGN KEY (Book\_ID) REFERENCES Books(Book\_ID),

    CONSTRAINT fk\_Member FOREIGN KEY (Member\_ID) REFERENCES Members(Member\_ID),

    CONSTRAINT fk\_Staff FOREIGN KEY (Staff\_ID) REFERENCES Staff(Staff\_ID)

)

CREATE TABLE Penalties(

Penalty\_ID INT IDENTITY(1,1) Primary Key,

Member\_ID INT,

Book\_ID INT,

Due\_Date DATETIME2,

Return\_Date DATETIME2,

Penalty\_Amount MONEY,

Penalty\_Status BIT

CONSTRAINT fk\_Penalty\_Member FOREIGN KEY (Member\_ID) REFERENCES Members(Member\_ID),

CONSTRAINT fk\_Penalty\_Book FOREIGN KEY (Book\_ID) REFERENCES Books(Book\_ID)

)

ALTER TABLE Positions ALTER COLUMN Position\_Name VARCHAR(40);

ALTER TABLE Staff ADD City\_ID TINYINT;

ALTER TABLE Staff ADD CONSTRAINT fk\_City FOREIGN KEY (City\_ID) REFERENCES Cities(City\_ID);

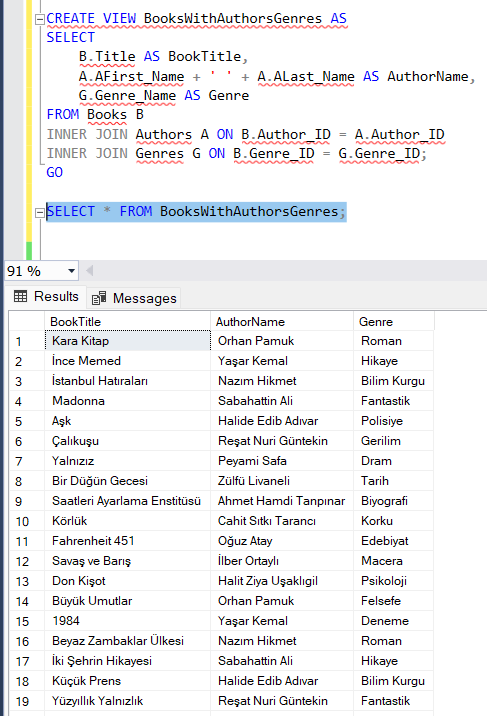
ALTER TABLE Staff DROP CONSTRAINT fk\_City;

ALTER TABLE Staff DROP COLUMN City\_ID;

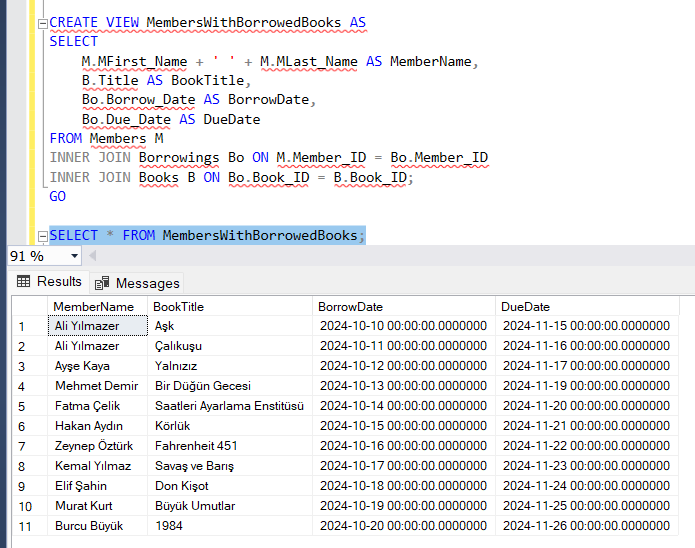
DROP DATABASE Kutuphane\_Yonetim\_Sistemi;

* **VIEW kullanımı**

*View 1: Türleri ve yazarları ile kitapların listesi*

****

*-- View 2: Üyelerin ve ödünç aldıkları kitapların listesi*

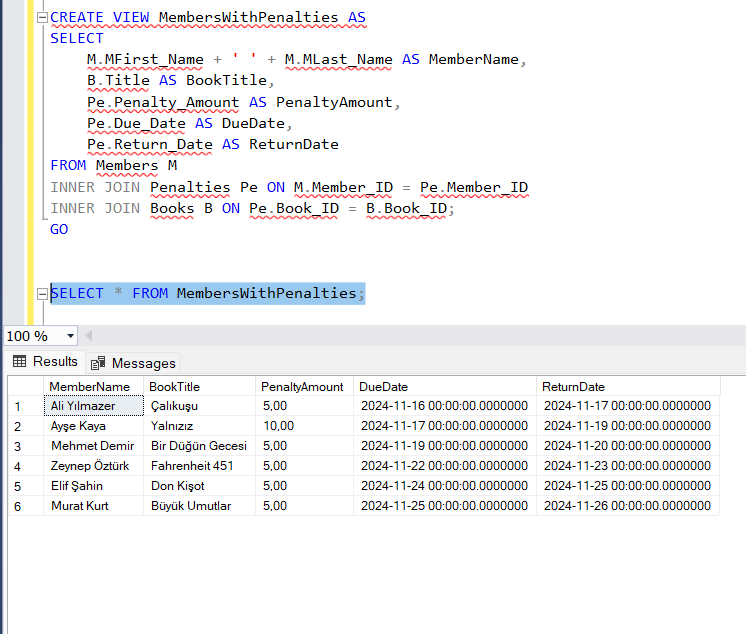
**

**--** *View 3: Pozisyonları ile personel*

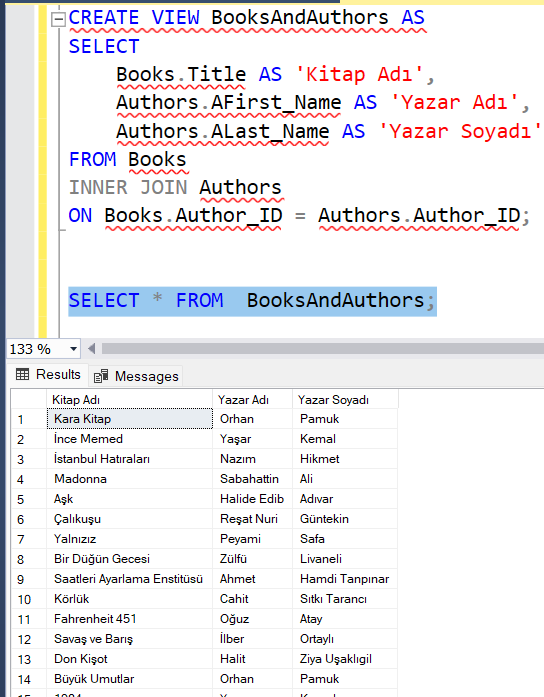
*metin, elektronik donanım, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu*

*-- View 4: Cezalı üyeler*

****

*-- View 5: Kitapların yazar isim soyisim bilgisi*

****

* **INDEX Kullanımı**

**1. Members Tablosu için Telefon Numarası İndeksi:**

CREATE INDEX idx\_members\_phone\_number ON Members(MPhone\_Number);

**2. Penalties Tablosu için Ceza Durumu ve Üye Kimliği İndeksi:**

CREATE INDEX idx\_penalties\_status\_member ON Penalties(Penalty\_Status, Member\_ID);

**3. Authors Tablosu için Yazar Adı ve Soyadı İndeksi:**

CREATE INDEX idx\_authors\_full\_name ON Authors(AFirst\_Name, ALast\_Name);

**4. Borrowings Tablosu için Üye ve Kitap İndeksi:**

CREATE INDEX idx\_borrowings\_member\_book ON Borrowings(Member\_ID, Book\_ID);

**5. Publishers Tablosu için Yayınevi Adı İndeksi:**

CREATE INDEX idx\_publishers\_name ON Publishers(Publisher\_Name);

* **Kitabın ödünç alınma, kitap durumunun güncellenmesi ve ödünç alma kaydının eklenmesini içeren transaction**

CREATE PROCEDURE BorrowBook

@BookID INT,

@MemberID INT,

@StaffID INT,

@BorrowDate DATETIME2,

@DueDate DATETIME2

AS

BEGIN

BEGIN TRANSACTION;

BEGIN TRY

     UPDATE Books

     SET Book\_Status = 0

     WHERE Book\_ID = @BookID;

     IF EXISTS (SELECT 1 FROM Books WHERE Book\_ID = @BookID AND Book\_Status = 0)

     BEGIN

         THROW 50001, 'Kitap zaten ödünçte.', 1;

     END

     INSERT INTO Borrowings (Book\_ID, Member\_ID, Borrow\_Date, Due\_Date, Staff\_ID)

     VALUES (@BookID, @MemberID, @BorrowDate, @DueDate, @StaffID);

     COMMIT TRANSACTION;

     PRINT 'Kitap ödünç alma işlemi başarılı bir şekilde tamamlandı.';

END TRY

BEGIN CATCH

     ROLLBACK TRANSACTION;

     PRINT 'Hata oluştu: İşlem geri alındı.';

     THROW;

END CATCH

END;

* **STORED PROCEDURE**

**1-Kitapları belirli bir türe göre listeleyen stored procedure kodu :**

CREATE PROCEDURE GetBooksByGenre

@genreName NVARCHAR(50)

AS BEGIN

 SELECT b.Title, b.Publicition\_Year, g.Genre\_Name

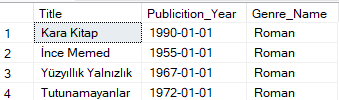
 FROM Books b JOIN Genres g

ON b.Genre\_ID = g.Genre\_ID

WHERE g.Genre\_Name = @genreName;

END

–EXEC GetBooksByGenre @genreName = 'Roman';

****

**2- Kitapların listesini ve yazar bilgilerini getiren Stored Procedure kodu:**

