

SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİTABANI YÖNETİM SİSTEMLERİ PROJE ÖDEVİ

AD: Elif

SOYAD: Uzunoğlu

ÖĞRENCİ NUMARASI: B251210379

E-posta: elif.uzunoglu1@gmail.com

Uygulamanın Tanıtımı:

Bu bir kütüphane yönetim sistemi uygulamasıdır. Amaç Kütüphane personelinin işini kolaylaştırmak, kitap ve üye takibini hızlandırmaktır. Üye ekleme, güncelleme, silme gibi birçok işlemi yerine getirir. Stored procedure ve trigger'larla otomatik işlemler yaptırıyoruz. Mesela ceza hesaplama, kopya ekleme gibi işleri veritabanı kendisi hallediyor. TC kimlik kontrolü, veri doğrulama, hata mesajları falan var. Bu sayede sistem korunuyor.

İş Kuralları:

Bir kişi personel ya da bir üye olabilir.

Üyeler öğrenci, öğretmen veya diğer türlerinden yalnızca biri olabilir.

Öğrencinin okuduğu okul , bölüm ve sınıf bilgisi olmalı.

Öğretmenin branşı , iş yeri bilgisi olmalı.

Kitap materyalleri bulunur.

Kişilerin adı, soyadı, telefon numarası, e-mail adresi bulunmalıdır.

Personel'in sicil numarası, çalıştığı pozisyon, ev adresi olmalıdır.

Üye'nin tür bilgisi, üye ID'si olmalıdır.

Materyallerin materyalNo, adı, yazarı, yayınevi, basım tarihi, stok bilgisi , baskı sayısı olmalı

Yazarın adı, soyadı , yazar ID olmalı.

Bir kitaba ait tür bilgisi (roman, polisiye, hikaye, şiir, felsefe) bulunmalıdır.

Bir kişinin en fazla bir üyeliği olabilir.

Bir kişi birden fazla kopya alabilir ancak bir kopya bir kişi tarafından alınabilir.

Bir kitabın birden çok yazarı olabilir ve bir yazarın birden çok kitabı olabilir.

Bir derginin birden çok yazarı olabilir ve bir yazarın birden çok dergisi olabilir.

Bir kitabın birden fazla türü olabilir ama bir tür en az bir kitaba ait olmalıdır.

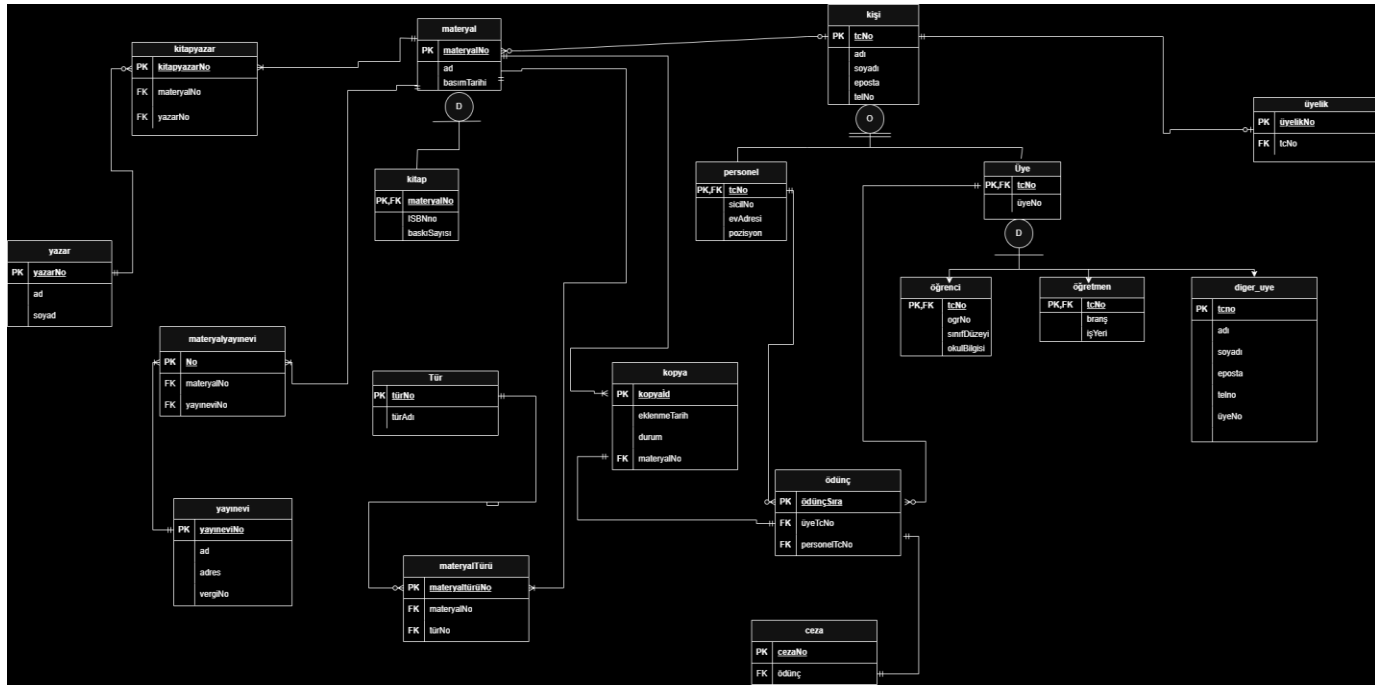
Bir kitap en az bir yayınevine ait olmalıdır ancak bir yayınevinin birden çok kitabı olabilir

Her kitaptan birden fazla kopya olabilir.

Bir üye aynı anda birden fazla kitap ödünç alabilir.

Ödünç süresi 14 gündür. iade geciktiğinde kullanıcı cezaya tabii tutulur.

Varlık Bağlantı Diyagramı:



İLİŞKİSEL ŞEMA:

Kişi(**tcNo:int** , adı:char , soyadı: varchar , eposta:varchar , telNo:varchar)

Personel(**tcNo:int**, sicilNo:int , evAdresi:varchar , pozisyon:varchar)

Üye(**tcNo: int**, üyeNo:int)

Öğrenci(**tcNo:int**, ogrNo:int, sınıfDüzeyi:int , okulBilgisi:varchar)

Öğretmen(**tcNo:int**, branş:char, işYeri:varchar)

Diğer_uye(**tcno:int** , adı:char, soyadı:char, eposta:char, telno:char , üyeNo:int)

Materyal(**materyalNo:int**, ad:char, basımTarihi:date)

Kitap(**materyalNo:int**, ISBNno:char, baskıSayısı:int)

Dergi(**materyalNo:int** , ciltNo:int)

materyalTürü(**materyalTürüNo:int** , materyalNo:int , türNo:int)

Tür(**türNo:int** , türAdı:char)

Yayınevi(**yayıneviNo:int**, ad:char, adres:char, vergiNo:int)

Materyalyayınevi(**no:int**, materyalNo:int , yayıneviNo:int)

Yazar(**yazarNo:int** , ad:char, soyad:char)

Kitapyazar(**kitapyazarNo:int** , materyalNo:int, yazarNo:int)

Üyelik(**üyelikNo:int**, tcNo:int)

Kopya(**kopyald**, eklenmeTarih:date ,durum:char, materyalNo:int)

ÖdünçAlma(**ödünçSıra:int**, üyeTcNo:int , personelTcNo:int)

Ceza(**cezaNo:int** , ödünç:int)

SQL İFADELERİ:

```
CREATE TABLE yazar (  
    yazarid SERIAL PRIMARY KEY,  
    ad VARCHAR(50) NOT NULL,  
    soyad VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE yayinevi (  
    yayinevino SERIAL PRIMARY KEY,  
    ad VARCHAR(100) UNIQUE NOT NULL,  
    adres VARCHAR(255),  
    vergino VARCHAR(20) UNIQUE  
);
```

```
CREATE TABLE tur (  
    turno SERIAL PRIMARY KEY,  
    turadi VARCHAR(50) UNIQUE NOT NULL  
);
```

```
CREATE TABLE kisi (  
    tcno CHAR(11) PRIMARY KEY,  
    adi VARCHAR(50) NOT NULL,  
    soyadi VARCHAR(50) NOT NULL,  
    eposta VARCHAR(100) UNIQUE,  
    telno VARCHAR(15)  
);
```

```
CREATE TABLE personel (  
    sicilno VARCHAR(20) UNIQUE NOT NULL,  
    evadres VARCHAR(255),  
    pozisyon VARCHAR(50),  
    PRIMARY KEY (tcno)  
    ) INHERITS (kisi);
```

```
CREATE TABLE uye (  
    uyeno VARCHAR(20) UNIQUE NOT NULL,  
    PRIMARY KEY (tcno)  
    ) INHERITS (kisi);
```

```
CREATE TABLE ogrenci (  
    ogrno VARCHAR(20) UNIQUE,  
    sinifduzeyi VARCHAR(20),  
    okulbilgisi VARCHAR(100),  
    PRIMARY KEY (tcno)  
) INHERITS (uye);
```

```
CREATE TABLE ogretmen (  
    brans VARCHAR(50),  
    isyeri VARCHAR(100),  
    PRIMARY KEY (tcno)  
) INHERITS (uye);
```

```
CREATE TABLE diger_uye (  
    gerekce VARCHAR(255),  
    gecerlilikbitistarihi DATE,  
    PRIMARY KEY (tcno)  
) INHERITS (uye);
```

```
CREATE TABLE uyelik (  
    uyelikid SERIAL PRIMARY KEY,  
    tcno CHAR(11) UNIQUE NOT NULL REFERENCES uye (tcno) ON DELETE CASCADE,  
    max_odunc_limit INT DEFAULT 5,  
    odunc_gun_suresi INT DEFAULT 15  
);
```

```
CREATE TABLE materyal (  
    materyalno SERIAL PRIMARY KEY,  
    ad VARCHAR(255) NOT NULL,  
    basimtarihi DATE  
);
```

```
CREATE TABLE kitap (  
    isbnno VARCHAR(20) UNIQUE,  
    baskisayisi INT,  
    PRIMARY KEY (materyalno)  
) INHERITS (materyal);
```

```
CREATE TABLE dergi (  

```

```
    ciltno VARCHAR(20),  
    PRIMARY KEY (materyalno)  
) INHERITS (materyal);
```

```
CREATE TABLE kitapyazar (  
    materyalno INT NOT NULL REFERENCES kitap (materyalno) ON DELETE CASCADE,  
    yazarid INT NOT NULL REFERENCES yazar (yazarid) ON DELETE CASCADE,  
    PRIMARY KEY (materyalno, yazarid)  
);
```

```
CREATE TABLE materyalyayinevi (  
    materyalno INT NOT NULL REFERENCES materyal (materyalno) ON DELETE CASCADE,  
    yayinevino INT NOT NULL REFERENCES yayinevi (yayinevino) ON DELETE CASCADE,  
    PRIMARY KEY (materyalno, yayinevino)  
);
```

```
CREATE TABLE materyaltur (  
    materyalno INT NOT NULL REFERENCES materyal (materyalno) ON DELETE CASCADE,  
    turno INT NOT NULL REFERENCES tur (turno) ON DELETE CASCADE,  
    PRIMARY KEY (materyalno, turno)  
);
```

```
CREATE TABLE kopya (  
    kopyaid SERIAL PRIMARY KEY,  
    materyalno INT NOT NULL REFERENCES materyal (materyalno) ON DELETE CASCADE,  
    rafno VARCHAR(10),  
    durum VARCHAR(20) DEFAULT 'Rafta' CHECK (durum IN ('Rafta', 'Ödünçte', 'Kayıp',  
'Bakımda')),  
    eklenmetarihi DATE DEFAULT CURRENT_DATE  
);
```

```
CREATE TABLE oduncalma (  
    oduncid SERIAL PRIMARY KEY,  
    kopyaid INT UNIQUE NOT NULL REFERENCES kopya (kopyaid),  
    uyetcno CHAR(11) NOT NULL REFERENCES uye (tcno),  
    personelcno CHAR(11) REFERENCES personel (tcno),  
    odunc_tarihi DATE DEFAULT CURRENT_DATE,  
    iade_tarihi_beklenen DATE NOT NULL  
);
```

```

CREATE TABLE ceza (
    cezaid SERIAL PRIMARY KEY,
    oduncid INT UNIQUE NOT NULL REFERENCES oduncalma (oduncid) ON DELETE CASCADE,
    iade_tarihi_gercek DATE,
    gecikme_gun INT,
    ceza_miktari DECIMAL(10, 2) DEFAULT 0.00,
    odeme_durumu BOOLEAN DEFAULT FALSE
);

```

```

CREATE OR REPLACE FUNCTION hesapla_uye_borc(p_tcno VARCHAR(11))
RETURNS DECIMAL(10,2)
LANGUAGE plpgsql
AS $$
DECLARE
    toplam_borc DECIMAL(10,2);
BEGIN
    SELECT COALESCE(SUM(ceza_miktari), 0)
    INTO toplam_borc
    FROM ceza c
    JOIN oduncalma o ON c.oduncid = o.oduncid
    WHERE o.uyetchno = p_tcno AND c.odeme_durumu = FALSE;

    RETURN toplam_borc;
END;
$$;

```

```

CREATE OR REPLACE FUNCTION kitap_stok_durumu(p_materyal_no INT)
RETURNS TABLE(
    toplam_kopya BIGINT,
    musait_kopya BIGINT,
    oduncte_kopya BIGINT,
    kayip_kopya BIGINT
)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY
    SELECT
        COUNT(*) as toplam_kopya,
        COUNT(*) FILTER (WHERE durum = 'Rafta') as musait_kopya,

```

```

COUNT(*) FILTER (WHERE durum = 'Ödünçte') as odunte_kopya,
COUNT(*) FILTER (WHERE durum = 'Kayıp') as kayip_kopya
FROM kopya
WHERE materyalno = p_materyal_no;
END;
$$;

```

```

CREATE OR REPLACE FUNCTION geciken_oduncler()
RETURNS TABLE(
    odunc_id INT,
    uye_ad_soyad VARCHAR,
    materyal_adi VARCHAR,
    gecikme_gun INT,
    potansiyel_ceza DECIMAL(10,2)
)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY
    SELECT
        o.oduncid,
        (u.adi || ' ' || u.soyadi)::VARCHAR,
        m.ad::VARCHAR,
        (CURRENT_DATE - o.iade_tarihi_beklenen)::INT,
        ((CURRENT_DATE - o.iade_tarihi_beklenen) * 2.0)::DECIMAL(10,2)
    FROM oduncalma o
    JOIN kopya k ON o.kopyaid = k.kopyaid
    JOIN materyal m ON k.materyalno = m.materyalno
    JOIN uye u ON o.uyetcno = u.tcno
    WHERE k.durum = 'Ödünçte'
    AND CURRENT_DATE > o.iade_tarihi_beklenen
    ORDER BY gecikme_gun DESC;
END;
$$;

```

```

CREATE OR REPLACE FUNCTION uye_ara(p_arama_terimi TEXT)
RETURNS TABLE(
    tcno CHAR(11),
    adi VARCHAR(50),
    soyadi VARCHAR(50),

```



```

eposta VARCHAR(100),
telno VARCHAR(15),
uyeno VARCHAR(20),
uye_tipi TEXT
)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY
    SELECT
        u.tcno,
        u.adi,
        u.soyadi,
        u.eposta,
        u.telno,
        u.uyeno,
        CASE
            WHEN og.tcno IS NOT NULL THEN 'Öğrenci'
            WHEN ot.tcno IS NOT NULL THEN 'Öğretmen'
            WHEN du.tcno IS NOT NULL THEN 'Diğer Üye'
            ELSE 'Üye'
        END::TEXT AS uye_tipi
    FROM uye u
    LEFT JOIN ogrenci og ON u.tcno = og.tcno
    LEFT JOIN ogretmen ot ON u.tcno = ot.tcno
    LEFT JOIN diger_uye du ON u.tcno = du.tcno
    WHERE
        u.adi ILIKE '%' || p_arama_terimi || '%' OR
        u.soyadi ILIKE '%' || p_arama_terimi || '%' OR
        u.tcno LIKE '%' || p_arama_terimi || '%' OR
        u.eposta ILIKE '%' || p_arama_terimi || '%' OR
        u.uyeno LIKE '%' || p_arama_terimi || '%'
    ORDER BY u.adi, u.soyadi;
END;
$$;

```

```

CREATE OR REPLACE FUNCTION uye_detay(p_tcno CHAR(11))
RETURNS TABLE(
    tcno CHAR(11),
    adi VARCHAR(50),

```

```

    soyadi VARCHAR(50),
    eposta VARCHAR(100),
    telno VARCHAR(15),
    uyeno VARCHAR(20),
    uye_tipi TEXT,
    aktif_odunc BIGINT,
    toplam_borc DECIMAL(10,2)
)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY
    SELECT
        u.tcno,
        u.adi,
        u.soyadi,
        u.eposta,
        u.telno,
        u.uyeno,
        CASE
            WHEN og.tcno IS NOT NULL THEN 'Öğrenci'
            WHEN ot.tcno IS NOT NULL THEN 'Öğretmen'
            WHEN du.tcno IS NOT NULL THEN 'Diğer Üye'
            ELSE 'Üye'
        END::TEXT AS uye_tipi,
        (SELECT COUNT(*)
         FROM oduncalma o
         JOIN kopya k ON o.kopyaid = k.kopyaid
         WHERE o.uyetcno = u.tcno AND k.durum = 'Ödünçte') AS aktif_odunc,
        hesapla_uye_borc(u.tcno) AS toplam_borc
    FROM uye u
    LEFT JOIN ogrenci og ON u.tcno = og.tcno
    LEFT JOIN ogretmen ot ON u.tcno = ot.tcno
    LEFT JOIN diger_uye du ON u.tcno = du.tcno
    WHERE u.tcno = p_tcno;
END;
$$;

CREATE OR REPLACE FUNCTION aktif_uyeler()
RETURNS TABLE(

```

```

    tcno CHAR(11),
    tam_ad VARCHAR(101),
    uye_tipi TEXT,
    aktif_odunc BIGINT,
    toplam_borc DECIMAL(10,2)
)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY
    SELECT
        u.tcno,
        (u.adi || ' ' || u.soyadi)::VARCHAR(101) AS tam_ad,
        CASE
            WHEN og.tcno IS NOT NULL THEN 'Öğrenci'
            WHEN ot.tcno IS NOT NULL THEN 'Öğretmen'
            WHEN du.tcno IS NOT NULL THEN 'Diğer Üye'
            ELSE 'Üye'
        END::TEXT AS uye_tipi,
        (SELECT COUNT(*)
         FROM oduncalma o
         JOIN kopya k ON o.kopyaid = k.kopyaid
         WHERE o.uyetcno = u.tcno AND k.durum = 'Ödünçte') AS aktif_odunc,
        hesapla_uye_borc(u.tcno) AS toplam_borc
    FROM uye u
    LEFT JOIN ogrenci og ON u.tcno = og.tcno
    LEFT JOIN ogretmen ot ON u.tcno = ot.tcno
    LEFT JOIN diger_uye du ON u.tcno = du.tcno
    WHERE EXISTS (
        SELECT 1 FROM oduncalma o
        JOIN kopya k ON o.kopyaid = k.kopyaid
        WHERE o.uyetcno = u.tcno AND k.durum = 'Ödünçte'
    )
    OR hesapla_uye_borc(u.tcno) > 0
    ORDER BY u.adi, u.soyadi;
END;
$$;

CREATE OR REPLACE FUNCTION borclu_uyeler()
RETURNS TABLE(

```

```

    tcno CHAR(11),
    tam_ad VARCHAR(101),
    uye_tipi TEXT,
    toplam_borc DECIMAL(10,2),
    odenmemis_ceza_sayisi BIGINT
)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY
    SELECT
        u.tcno,
        (u.adi || ' ' || u.soyadi)::VARCHAR(101) AS tam_ad,
        CASE
            WHEN og.tcno IS NOT NULL THEN 'Öğrenci'
            WHEN ot.tcno IS NOT NULL THEN 'Öğretmen'
            WHEN du.tcno IS NOT NULL THEN 'Diğer Üye'
            ELSE 'Üye'
        END::TEXT AS uye_tipi,
        hesapla_uye_borc(u.tcno) AS toplam_borc,
        (SELECT COUNT(*))
        FROM ceza c
        JOIN oduncalma o ON c.oduncid = o.oduncid
        WHERE o.uyetcno = u.tcno AND c.odeme_durumu = FALSE) AS odenmemis_ceza_sayisi
    FROM uye u
    LEFT JOIN ogrenci og ON u.tcno = og.tcno
    LEFT JOIN ogretmen ot ON u.tcno = ot.tcno
    LEFT JOIN diger_uye du ON u.tcno = du.tcno
    WHERE hesapla_uye_borc(u.tcno) > 0
    ORDER BY toplam_borc DESC;
END;
$$;

CREATE OR REPLACE PROCEDURE sp_odunc_ver(
    p_uye_tcno CHAR(11),
    p_kopya_id INT,
    p_personel_tcno CHAR(11) DEFAULT NULL
)
LANGUAGE plpgsql
AS $$

```

```

DECLARE
    v_max_limit INT;
    v_aktif_odunc INT;
    v_toplam_borc DECIMAL(10,2);
    v_odunc_suresi INT;
    v_iade_tarihi DATE;
    v_kopya_durum VARCHAR(20);
BEGIN
    SELECT durum INTO v_kopya_durum
    FROM kopya WHERE kopyaid = p_kopya_id;

    IF v_kopya_durum IS NULL THEN
        RAISE EXCEPTION 'Kopya bulunamadı!';
    END IF;

    IF v_kopya_durum != 'Rafta' THEN
        RAISE EXCEPTION 'Bu kopya şu anda müsait değil! Durum: %', v_kopya_durum;
    END IF;

    SELECT max_odunc_limit, odunc_gun_suresi
    INTO v_max_limit, v_odunc_suresi
    FROM uyelik WHERE tcno = p_uye_tcno;

    IF v_max_limit IS NULL THEN
        RAISE EXCEPTION 'Üyelik bilgisi bulunamadı!';
    END IF;

    SELECT COUNT(*)
    INTO v_aktif_odunc
    FROM oduncalma o
    JOIN kopya k ON o.kopyaid = k.kopyaid
    WHERE o.uyetchno = p_uye_tcno AND k.durum = 'Ödünçte';

    IF v_aktif_odunc >= v_max_limit THEN
        RAISE EXCEPTION 'Ödünç limiti dolmuş! (Limit: %, Aktif: %)', v_max_limit, v_aktif_odunc;
    END IF;

    v_toplam_borc := hesapla_uye_borc(p_uye_tcno);

    IF v_toplam_borc > 0 THEN

```

```
RAISE EXCEPTION 'Ödenmemiş ceza var: % TL', v_toplam_borc;
END IF;
```

```
v_iade_tarihi := CURRENT_DATE + v_odunc_suresi;
```

```
INSERT INTO oduncalma (kopyaid, uyetcno, personelcno, odunc_tarihi,
iade_tarihi_beklenen)
VALUES (p_kopya_id, p_uye_tcno, p_personel_tcno, CURRENT_DATE, v_iade_tarihi);
```

```
UPDATE kopya SET durum = 'Ödünçte' WHERE kopyaid = p_kopya_id;
```

```
RAISE NOTICE 'Ödünç verme başarılı! İade tarihi: %', v_iade_tarihi;
END;
$$;
```

```
CREATE OR REPLACE PROCEDURE sp_iade_al(p_odunc_id INT)
LANGUAGE plpgsql
AS $$
```

```
DECLARE
```

```
v_kopya_id INT;
v_iade_tarihi_beklenen DATE;
v_gecikme_gun INT;
v_ceza_miktari DECIMAL(10,2);
```

```
BEGIN
```

```
SELECT kopyaid, iade_tarihi_beklenen
INTO v_kopya_id, v_iade_tarihi_beklenen
FROM oduncalma
WHERE oduncid = p_odunc_id;
```

```
IF v_kopya_id IS NULL THEN
RAISE EXCEPTION 'Ödünç kaydı bulunamadı!';
END IF;
```

```
v_gecikme_gun := CURRENT_DATE - v_iade_tarihi_beklenen;
```

```
IF v_gecikme_gun > 0 THEN
v_ceza_miktari := v_gecikme_gun * 2.0;
```

```
INSERT INTO ceza (oduncid, iade_tarihi_gercek, gecikme_gun, ceza_miktari,
odeme_durumu)
```

```

VALUES (p_odunc_id, CURRENT_DATE, v_gecikme_gun, v_ceza_miktari, FALSE);

RAISE NOTICE 'İade alındı. % gün gecikme! Ceza: % TL', v_gecikme_gun, v_ceza_miktari;
ELSE
RAISE NOTICE 'İade alındı. Gecikme yok.';
END IF;

UPDATE kopya SET durum = 'Rafta' WHERE kopyaid = v_kopya_id;
END;
$$;

CREATE OR REPLACE PROCEDURE sp_toplu_ceza_ode(p_uye_tcn CHAR(11))
LANGUAGE plpgsql
AS $$
DECLARE
    v_toplam_borc DECIMAL(10,2);
    v_odenen_ceza_sayisi INT;
BEGIN
    v_toplam_borc := hesapla_uye_borc(p_uye_tcn);

    IF v_toplam_borc = 0 THEN
        RAISE NOTICE 'Ödenmemiş ceza bulunmamaktadır.';
        RETURN;
    END IF;

    UPDATE ceza c
    SET odeme_durumu = TRUE
    FROM oduncalma o
    WHERE c.oduncid = o.oduncid
    AND o.uyetchno = p_uye_tcn
    AND c.odeme_durumu = FALSE;

    GET DIAGNOSTICS v_odenen_ceza_sayisi = ROW_COUNT;

    RAISE NOTICE '% adet ceza ödendi. Toplam: % TL', v_odenen_ceza_sayisi, v_toplam_borc;
END;
$$;

CREATE OR REPLACE PROCEDURE sp_kopya_ekle(
    p_materyal_no INT,

```

```

    p_kopya_sayisi INT DEFAULT 1,
    p_raf_prefix VARCHAR(10) DEFAULT 'R'
)
LANGUAGE plpgsql
AS $$
DECLARE
    i INT;
    v_materyal_adi VARCHAR(255);
BEGIN
    SELECT ad INTO v_materyal_adi
    FROM materyal WHERE materyalno = p_materyal_no;

    IF v_materyal_adi IS NULL THEN
        RAISE EXCEPTION 'Materyal bulunamadı!';
    END IF;

    FOR i IN 1..p_kopya_sayisi LOOP
        INSERT INTO kopya (materyalno, rafno, durum, eklenmetarihi)
        VALUES (
            p_materyal_no,
            p_raf_prefix || '-' || p_materyal_no || '-' || (SELECT COUNT(*) + 1 FROM kopya
WHERE materyalno = p_materyal_no),
            'Rafta',
            CURRENT_DATE
        );
    END LOOP;

    RAISE NOTICE '% adet kopya eklendi: %', p_kopya_sayisi, v_materyal_adi;
END;
$$;

CREATE OR REPLACE PROCEDURE sp_ogrenci_ekle(
    p_tcno VARCHAR,
    p_ad VARCHAR,
    p_soyad VARCHAR,
    p_eposta VARCHAR,
    p_telno VARCHAR,
    p_ogrno VARCHAR,
    p_sinifduzeyi VARCHAR,
    p_okulbilgisi VARCHAR

```



```

)
LANGUAGE plpgsql
AS $$
BEGIN
    INSERT INTO uye (tcno, adi, soyadi, eposta, telno, uyeno)
    VALUES (p_tcno, p_ad, p_soyad, p_eposta, p_telno, p_ogrno);

    INSERT INTO ogrenci (tcno, adi, soyadi, eposta, telno, uyeno, ogrno, sinifduzeyi, okulbilgisi)
    VALUES (p_tcno, p_ad, p_soyad, p_eposta, p_telno, p_ogrno, p_ogrno, p_sinifduzeyi,
    p_okulbilgisi);

    RAISE NOTICE 'Öğrenci kaydı oluşturuldu: % %', p_ad, p_soyad;
EXCEPTION
    WHEN unique_violation THEN
        RAISE EXCEPTION 'Bu TC numarası ile kayıtlı bir üye zaten var';
    WHEN OTHERS THEN
        RAISE;
END;
$$;

```

```

CREATE OR REPLACE PROCEDURE sp_ogretmen_ekle(
    p_tcno VARCHAR,
    p_ad VARCHAR,
    p_soyad VARCHAR,
    p_eposta VARCHAR,
    p_telno VARCHAR,
    p_brans VARCHAR,
    p_isyeri VARCHAR
)
LANGUAGE plpgsql
AS $$
DECLARE
    v_uyeno VARCHAR;
BEGIN
    v_uyeno := 'OGR-' || p_tcno;

    INSERT INTO uye (tcno, adi, soyadi, eposta, telno, uyeno)
    VALUES (p_tcno, p_ad, p_soyad, p_eposta, p_telno, v_uyeno);

    INSERT INTO ogretmen (tcno, adi, soyadi, eposta, telno, uyeno, brans, isyeri)

```

```

VALUES (p_tcnno, p_ad, p_soyad, p_eposta, p_telno, v_uyeno, p_brans, p_isyeri);

RAISE NOTICE 'Öğretmen kaydı oluşturuldu: % %', p_ad, p_soyad;
EXCEPTION
    WHEN unique_violation THEN
        RAISE EXCEPTION 'Bu TC numarası ile kayıtlı bir üye zaten var';
    WHEN OTHERS THEN
        RAISE;
END;
$$;

CREATE OR REPLACE PROCEDURE sp_diger_uye_ekle(
    p_tcnno VARCHAR,
    p_ad VARCHAR,
    p_soyad VARCHAR,
    p_eposta VARCHAR,
    p_telno VARCHAR,
    p_gerekce TEXT,
    p_gecerlilik DATE
)
LANGUAGE plpgsql
AS $$
DECLARE
    v_uyeno VARCHAR;
BEGIN
    v_uyeno := 'DGR-' || p_tcnno;

    INSERT INTO uye (tcno, adi, soyadi, eposta, telno, uyeno)
    VALUES (p_tcnno, p_ad, p_soyad, p_eposta, p_telno, v_uyeno);

    INSERT INTO diger_uye (tcno, adi, soyadi, eposta, telno, uyeno, gerekce,
    gecerlilikbitistarihi)
    VALUES (p_tcnno, p_ad, p_soyad, p_eposta, p_telno, v_uyeno, p_gerekce, p_gecerlilik);

    RAISE NOTICE 'Diğer üye kaydı oluşturuldu: % %', p_ad, p_soyad;
EXCEPTION
    WHEN unique_violation THEN
        RAISE EXCEPTION 'Bu TC numarası ile kayıtlı bir üye zaten var';
    WHEN OTHERS THEN
        RAISE;

```

END;

\$\$;

CREATE OR REPLACE PROCEDURE sp_uye_guncelle(

 p_tcnno CHAR(11),

 p_ad VARCHAR(50),

 p_soyad VARCHAR(50),

 p_eposta VARCHAR(100),

 p_telno VARCHAR(15)

)

LANGUAGE plpgsql

AS \$\$

BEGIN

 UPDATE uye

 SET

 adi = p_ad,

 soyadi = p_soyad,

 eposta = p_eposta,

 telno = p_telno

 WHERE tcno = p_tcnno;

 IF NOT FOUND THEN

 RAISE EXCEPTION 'Üye bulunamadı: %', p_tcnno;

 END IF;

 RAISE NOTICE 'Üye güncellendi: % %', p_ad, p_soyad;

END;

\$\$;

CREATE OR REPLACE PROCEDURE sp_uye_sil(p_tcnno CHAR(11))

LANGUAGE plpgsql

AS \$\$

DECLARE

 v_uye_adi VARCHAR(101);

BEGIN

 SELECT adi || ' ' || soyadi INTO v_uye_adi

 FROM uye WHERE tcno = p_tcnno;

 IF v_uye_adi IS NULL THEN

 RAISE EXCEPTION 'Üye bulunamadı: %', p_tcnno;

END IF;

DELETE FROM uye WHERE tcno = p_tcno;

RAISE NOTICE 'Üye silindi: %', v_uye_adi;

END;

\$\$;

CREATE OR REPLACE FUNCTION trg_kopya_silme_koruma()

RETURNS TRIGGER

LANGUAGE plpgsql

AS \$\$

BEGIN

IF OLD.durum = 'Ödünçte' THEN

RAISE EXCEPTION 'Ödünçte olan kopya silinemez! Kopya ID: %', OLD.kopyaid;

END IF;

RETURN OLD;

END;

\$\$;

CREATE TRIGGER kopya_silme_trigger

BEFORE DELETE ON kopya

FOR EACH ROW

EXECUTE FUNCTION trg_kopya_silme_koruma();

CREATE OR REPLACE FUNCTION trg_uyelik_otomatik_olustur()

RETURNS TRIGGER

LANGUAGE plpgsql

AS \$\$

BEGIN

INSERT INTO uyelik (tcno, max_odunc_limit, odunc_gun_suresi)

VALUES (NEW.tcno, 5, 15)

ON CONFLICT (tcno) DO NOTHING;

RETURN NEW;

END;

\$\$;

CREATE TRIGGER uyelik_olustur_trigger

AFTER INSERT ON uye

FOR EACH ROW

```
EXECUTE FUNCTION trg_uyelik_otomatik_olustur();
```

```
CREATE OR REPLACE FUNCTION trg_uye_silme_koruma()
```

```
RETURNS TRIGGER
```

```
LANGUAGE plpgsql
```

```
AS $$
```

```
DECLARE
```

```
    v_aktif_odunc INT;
```

```
    v_odenmemis_borc DECIMAL(10,2);
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO v_aktif_odunc
```

```
    FROM oduncalma o
```

```
    JOIN kopya k ON o.kopyaid = k.kopyaid
```

```
    WHERE o.uyetcno = OLD.tcno AND k.durum = 'Ödünçte';
```

```
    IF v_aktif_odunc > 0 THEN
```

```
        RAISE EXCEPTION 'Üyenin % adet aktif ödöncü var, silinemez!', v_aktif_odunc;
```

```
    END IF;
```

```
    v_odenmemis_borc := hesapla_uye_borc(OLD.tcno);
```

```
    IF v_odenmemis_borc > 0 THEN
```

```
        RAISE EXCEPTION 'Üyenin % TL ödenmemiş borcu var, silinemez!', v_odenmemis_borc;
```

```
    END IF;
```

```
    RETURN OLD;
```

```
END;
```

```
$$;
```

```
CREATE TRIGGER uye_silme_koruma_trigger
```

```
BEFORE DELETE ON uye
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION trg_uye_silme_koruma();
```

```
CREATE OR REPLACE FUNCTION trg_tcno_kontrol()
```

```
RETURNS TRIGGER
```

```
LANGUAGE plpgsql
```

```
AS $$
```

```
BEGIN
```

```
    IF LENGTH(NEW.tcno) != 11 THEN
```

```
RAISE EXCEPTION 'TC Kimlik No 11 haneli olmalıdır!';  
END IF;
```

```
IF NEW.tcno !~ '^[0-9]{11}$' THEN  
    RAISE EXCEPTION 'TC Kimlik No sadece rakam içermelidir!';  
END IF;
```

```
IF LEFT(NEW.tcno, 1) = '0' THEN  
RAISE EXCEPTION 'TC Kimlik No 0 ile başlayamaz!';  
END IF;
```

```
RETURN NEW;  
END;  
$$;
```

```
CREATE TRIGGER tcno_kontrol_trigger  
BEFORE INSERT OR UPDATE ON kisi  
FOR EACH ROW  
EXECUTE FUNCTION trg_tcno_kontrol();
```

Prosedürler:

sp_odunc_ver - Ödünç verme işlemi (limit ve borç kontrolü ile)

sp_iade_al - İade alma ve gecikme cezası hesaplama

sp_toplu_ceza_ode - Toplu ceza ödeme işlemi

sp_kopya_ekle - Materyale yeni kopya ekleme

Triggerlar:

kopya_silme_trigger - Ödünçte olan kopya silinmesini engeller

uyelik_olustur_trigger - Yeni üyeye otomatik üyelik oluşturur

uye_silme_koruma_trigger - Borçlu/aktif ödünçlü üye silinmesini engeller

tcno_kontrol_trigger - TC Kimlik No formatını doğrular

Fonksiyonlar:

hesapla_uye_borc- Bir üyenin ödenmemiş toplam ceza borcunu hesaplar.

kitap_stok_durumu- Bir materyalin tüm kopyalarının durumunu özetler.

geciken_oduncler- Tüm gecikmiş ödünç kayıtlarını listeler.

uye_ara- Üyeleri ad, soyad, TC No, e-posta veya üye numarasına göre arar.

uye_detay- Bir üyenin detaylı bilgilerini ve hesap durumunu gösterir.

aktif_uyeler- Aktif ödöncü olan veya borcu bulunan tüm üyeleri listeler.

borclu_uyeler- Ödenmemiş cezası olan üyeleri listeler.

Kütüphane Yönetim Sistemi

+ Yeni Kayıt Ekle Arama & Güncelleme Kitap Yönetimi Ödünç İşlemleri Personel Yönetimi

+ YENİ ÜYE EKLEME

Üye Türü: ☐ Öğrenci ☒ Öğretmen ☐ Diğer Üye

TC Kimlik No *

4444444444

Ad *

Ali

Soyad *

Ateş

E-posta

alates@gmail.com

Telefon

543789230

Branş *

Bilgisayar mühendisliği

İş Yeri

sakarya universitesi

ÜYE KAYDI YAP

Başarılı

Öğretmen başarıyla eklendi:
Ali Ateş

Tamam

Kütüphane Yönetim Sistemi

+ Yeni Kayıt Ekle Arama & Güncelleme Kitap Yönetimi Ödünç İşlemleri Personel Yönetimi

KAYITLI ÜYELER VE ARAMA

Adı	Soyadı	Üye No	TC No
Ali	Tekin	UYE00018	95555555501
Ali	Buz	13	44444444444
Aylin	Kurt	UYE00015	98765432105
ayşe	tekin	UYE00024	12345678910
Ayşe	Demir	UYE00002	12345678902
Burak	Şahin	UYE00007	12345678907
Can	Özkan	UYE00005	12345678905
Deniz	Koç	UYE00010	12345678910
Elif	Anılan	UYE00006	12345678906
Emre	Aydın	UYE00009	12345678909
Fatma	Öztürk	UYE00011	98765432101
Gül	Acar	UYE00019	55555555502
Hacer	Güneş	UYE00021	55555555504
Hasan	Karaca	UYE00012	98765432102
İbrahim	Aksoy	UYE00016	98765432106
Kemal	Bulut	UYE00020	55555555503
Mehmet	Kaya	UYE00003	12345678903
Merve	Yaruz	UYE00017	98765432107
Mustafa	Erdoğan	UYE00014	98765432104
Osman	Taş	UYE00022	55555555505
Selin	Yıldız	UYE00008	12345678908
Serçi	Polat	UYE00013	98765432103
Zeynep	Çelik	UYE00004	12345678904

Listeyi Yenile

Detay Göster Üye Ara

Seçili Kaydı Güncelle / Sil

TC No: 44444444444

Ad: Ali

Soyadı: Buz

GÜNCELLE SİL

Kütüphane Yönetim Sistemi

+ Yeni Kayıt Ekle Arama & Güncelleme Kitap Yönetimi Ödünç İşlemleri Personel Yönetimi

KAYITLI ÜYELER VE ARAMA

Adı	Soyadı	Üye No	TC No
Ali	Tekin	UYE00018	95555555501
Ali	Ateş	13	44444444444
Aylin	Kurt	UYE00015	98765432105
ayşe	tekin	UYE00024	12345678910
Ayşe	Demir	UYE00002	12345678902
Burak	Şahin	UYE00007	12345678907
Can	Özkan	UYE00005	12345678905
Deniz	Koç	UYE00010	12345678910
Elif	Anılan	UYE00006	12345678906
Emre	Aydın	UYE00009	12345678909
Fatma	Öztürk	UYE00011	98765432101
Gül	Acar	UYE00019	55555555502
Hacer	Güneş	UYE00021	55555555504
Hasan	Karaca	UYE00012	98765432102
İbrahim	Aksoy	UYE00016	98765432106
Kemal	Bulut	UYE00020	55555555503
Mehmet	Kaya	UYE00003	12345678903
Merve	Yaruz	UYE00017	98765432107
Mustafa	Erdoğan	UYE00014	98765432104
Osman	Taş	UYE00022	55555555505
Selin	Yıldız	UYE00008	12345678908
Serçi	Polat	UYE00013	98765432103
Zeynep	Çelik	UYE00004	12345678904

Üye Ara

Ad

Ateş

Soyad

13

Ara

Kütüphane Yönetim Sistemi

Yeni Kayıt Ekle

Arama & Güncelleme

Kıtap Yönetimi

Ödünç İşlemleri

Personel Yönetimi

KAYITLI ÜYELER VE ARAMA

Adı	Soyadı	Üye No	TC No
Ali	Tekin	UYE00018	5555555501
Ali	Ateş	12	4444444444
Aylin	Kurt	UYE00015	98765432105
Ayşe	tekin	UYE00024	12345678910
Ayşe	Demir	UYE00002	12345678902
Burak	Şahin	UYE00007	12345678907
Can	Özkan	UYE00005	12345678905
Deniz	Koç	UYE00010	12345678910
Elif	Arslan	UYE00006	12345678906
Emre	Aydın	UYE00009	12345678909
Fatma	Öztürk	UYE00011	98765432101
Gül	Acar	UYE00019	5555555502
Hacer	Güneş	UYE00021	5555555504
Hasan	Karaca	UYE00012	98765432102
İbrahim	Aksoy	UYE00016	98765432106
Kemal	Bulut	UYE00020	5555555503
Mehmet	Kaya	UYE00003	12345678903
Merve	Yavuz	UYE00017	98765432107
Mustafa	Erdoğan	UYE00014	98765432104
Osman	Taş	UYE00022	5555555505
Selin	Yıldız	UYE00008	12345678908
Seydi	Polat	UYE00013	98765432103
Zeynep	Çelik	UYE00004	12345678904

Listeyi Yenile

Detay Göster

Üye Ara

Seçili Kayıtları Güncelle / Sil

TC No: 4444444444

Adı: Ali

Soyadı: Ateş

GÜNCELLE

SİL

Kütüphane Yönetim Sistemi

Yeni Kayıt Ekle

Arama & Güncelleme

Kıtap Yönetimi

Ödünç İşlemleri

Personel Yönetimi

KAYITLI ÜYELER VE ARAMA

Adı	Soyadı	Üye No	TC No
Ali	Tekin	UYE00018	5555555501
Ali	Buz	12	4444444444
Aylin	Kurt	UYE00015	98765432105
Ayşe	tekin	UYE00024	12345678910
Ayşe	Demir	UYE00002	12345678902
Burak	Şahin	UYE00007	12345678907
Can	Özkan	UYE00005	12345678905
Deniz	Koç	UYE00010	12345678910
Elif	Arslan	UYE00006	12345678906
Emre	Aydın	UYE00009	12345678909
Fatma	Öztürk	UYE00011	98765432101
Gül	Acar	UYE00019	5555555502
Hacer	Güneş	UYE00021	5555555504
Hasan	Karaca	UYE00012	98765432102
İbrahim	Aksoy	UYE00016	98765432106
Kemal	Bulut	UYE00020	5555555503
Mehmet	Kaya	UYE00003	12345678903
Merve	Yavuz	UYE00017	98765432107
Mustafa	Erdoğan	UYE00014	98765432104
Osman	Taş	UYE00022	5555555505
Selin	Yıldız	UYE00008	12345678908
Seydi	Polat	UYE00013	98765432103
Zeynep	Çelik	UYE00004	12345678904

Listeyi Yenile

Detay Göster

Üye Ara

Seçili Kayıtları Güncelle / Sil

TC No: 4444444444

Adı: Ali

Soyadı: Buz

GÜNCELLE

SİL

Onay

Üye silinsin mi?

Evet

Hayır

Github linki: <https://github.com/elifuzngluu/odev.git>