

```

Create table SALON (
salon_id number GENERATED ALWAYS AS IDENTITY PRIMARY KEY ,
salonlar varchar(10)
)
Create table IL (
il_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
il VARCHAR(30)
)
Create table ILCE (
ilce_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY ,
ilce varchar2(30),
il_id number,
CONSTRAINT fk_il_id FOREIGN KEY (il_id) REFERENCES IL(il_id)
)
Create table GAZETE_AD (
gazete_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY ,
gazete_ad varchar(50)
)
Create table TUR(
tur_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY ,
tur_ad varchar(50)
)

```

Öncelikle foregn key barındırmayan tabloları oluşturduk çünkü oradaki primary keyleri referans alan tablolarımız var

```

Create table YAZAR(
yazar_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
yazar_ad VARCHAR2(30),
yazar_soyad VARCHAR2(30),
cinsiyet varchar2(7)
);
Create table KITAP(
kitap_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
kitap_ad VARCHAR(100),
kitap_sayisi number,
salon_id number,
tur_id number,
CONSTRAINT fk_salon_id FOREIGN KEY (salon_id) REFERENCES SALON(salon_id),
CONSTRAINT fk_tur_id FOREIGN KEY (tur_id) REFERENCES TUR(tur_id)
);
CREATE TABLE DERGI (
    dergi_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY ,
    dergi_ad VARCHAR2(100),
    tur_id NUMBER,
    salon_id NUMBER,
    CONSTRAINT fk_salon_id_dergi FOREIGN KEY (salon_id) REFERENCES SALON(salon_id),
    CONSTRAINT fk_tur_id_dergi FOREIGN KEY (tur_id) REFERENCES TUR(tur_id)
)

```

Yazar, kitap ve dergi tablolarını oluşturduk burada yazarla kitap ve yazarla dergi çok çok ilişkide olacaklar hareket tablolarını eklediğim kısım aşağıda
kitap tablosunda salon_id foreign key kitabın hangi salonda olduğunu söylüyor
tür de kitabın türünü söylüyor tür tablosunu oluşturmada kitabın türünü söyleyemeyiz

```
47 CREATE TABLE Yazar_Kitap (  
48     yazar_id NUMBER,  
49     kitap_id NUMBER,  
50     PRIMARY KEY (yazar_id, kitap_id),  
51     FOREIGN KEY (yazar_id) REFERENCES Yazar(yazar_id),  
52     FOREIGN KEY (kitap_id) REFERENCES Kitap(kitap_id)  
53 );  
54 CREATE TABLE Yazar_Dergi (  
55     yazar_id NUMBER,  
56     dergi_id NUMBER,  
57     PRIMARY KEY (yazar_id, dergi_id),  
58     FOREIGN KEY (yazar_id) REFERENCES YAZAR(yazar_id),  
59     FOREIGN KEY (dergi_id) REFERENCES DERGI(dergi_id)  
60 );  
61  
62 CREATE TABLE UYE (  
63     uye_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
64     uye_ad VARCHAR(40),  
65     uye_soyad VARCHAR(40),  
66     kitap_id number,  
67     uye_tc VARCHAR(11),  
68     uye_tel VARCHAR(11),  
69     il_id NUMBER,  
70     ilce_id NUMBER,  
71     CONSTRAINT fk_kitap_id_uye FOREIGN KEY (kitap_id) REFERENCES KITAP(kitap_id),  
72     CONSTRAINT fk_il_id_uye FOREIGN KEY (il_id) REFERENCES IL(il_id),  
73     CONSTRAINT fk_ilce_id_uye FOREIGN KEY (ilce_id) REFERENCES ILCE(ilce_id)  
74 )
```

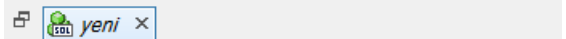
Burada da hareket tablolarını oluşturdu ve üye tablosunu oluşturduk üye tablosunda uyeler kitap alabildikleri için kitap ve üye tablosu arasında triggerlar oluşturacağız

```
,  
CREATE TABLE GAZETE(  
gazet_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
yillar VARCHAR2(4),  
gazete_id number,  
salon_id number ,  
CONSTRAINT fk_gazete_id_gaz FOREIGN KEY (gazete_id) REFERENCES GAZETE_AD(gazete_id),  
CONSTRAINT fk_salon_id_gaz FOREIGN KEY (salon_id) REFERENCES SALON(salon_id)  
)
```

Buda ekran çıktısı

Oracle SQL Developer : yeni

File Edit View Navigate Run Source Team Toc



Worksheet Query Builder

GO);

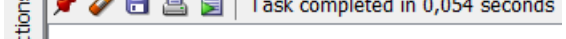
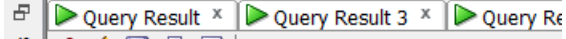


Table IL created.

Table ILCE created.

Table GAZETE_AD created.

Table TUR created.

Table YAZAR created.

Table KITAP created.

Table DERGI created.

Table YAZAR_KITAP created.

Table YAZAR_DERGI created.

Table UYE created.

Table GAZETE created.

VERİ EKLEME PAKET VE PROSEDÜR OLUŞTURMA VE VERİ EKLEME

```

create or replace NONEDITIONABLE PACKAGE veri_ekleme AS
    PROCEDURE ekle_salon(p_salonlar VARCHAR2);
    PROCEDURE ekle_il(p_il VARCHAR2);
    PROCEDURE ekle_ilce(p_ilce VARCHAR2, p_il_id NUMBER);
    PROCEDURE ekle_gazete_ad(p_gazete_ad VARCHAR2);
    PROCEDURE ekle_tur(p_tur_ad VARCHAR2);
    PROCEDURE ekle_yazar(p_yazar_ad VARCHAR2, p_yazar_soyad VARCHAR2, p_cinsiyet VARCHAR2);
    PROCEDURE ekle_kitap(p_kitap_ad VARCHAR2, p_kitap_sayisi NUMBER, p_salon_id NUMBER, p_tur_id NUMBER);
    PROCEDURE ekle_dergi(p_dergi_ad VARCHAR2, p_tur_id NUMBER, p_salon_id NUMBER);
    PROCEDURE ekle_yazar_kitap(p_yazar_id NUMBER, p_kitap_id NUMBER);
    PROCEDURE ekle_yazar_dergi(p_yazar_id NUMBER, p_dergi_id NUMBER);
    PROCEDURE ekle_uye(p_uye_ad VARCHAR2, p_uye_soyad VARCHAR2, p_kitap_id NUMBER,
        p_uye_tc VARCHAR2, p_uye_tel VARCHAR2, p_il_id NUMBER, p_ilce_id NUMBER);
    PROCEDURE ekle_gazete(p_yillar VARCHAR2, p_gazete_id NUMBER, p_salon_id NUMBER);
END veri_ekleme;

```

```

PROCEDURE ekle_tur(p_tur_ad VARCHAR2) AS
BEGIN
    INSERT INTO TUR (tur_ad) VALUES (p_tur_ad);
    DBMS_OUTPUT.PUT_LINE('Tür başarıyla eklendi: ' || p_tur_ad);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);
END ekle_tur;

PROCEDURE ekle_yazar(p_yazar_ad VARCHAR2, p_yazar_soyad VARCHAR2, p_cinsiyet VARCHAR2) AS
BEGIN
    INSERT INTO YAZAR (yazar_ad, yazar_soyad, cinsiyet) VALUES (p_yazar_ad, p_yazar_soyad, p_cinsiyet);
    DBMS_OUTPUT.PUT_LINE('Yazar başarıyla eklendi: ' || p_yazar_ad || ' ' || p_yazar_soyad);
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Bu yazar zaten mevcut.');
```

CodeErrorsProfilesGrantsDetailsReferencesDependencies

↑

create or replace NONEDITIONABLE PACKAGE BODY veri_ekleme AS

↑

PROCEDURE ekle_salon(p_salonlar VARCHAR2) AS

BEGIN

INSERT INTO SALON (salonlar) VALUES (p_salonlar);

DBMS_OUTPUT.PUT_LINE('Salon başarıyla eklendi: ' || p_salonlar);

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);

END ekle_salon;

↑

PROCEDURE ekle_il(p_il VARCHAR2) AS

BEGIN

INSERT INTO IL (il) VALUES (p_il);

DBMS_OUTPUT.PUT_LINE('İl başarıyla eklendi: ' || p_il);

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);

END ekle_il;

↑

PROCEDURE ekle_ilce(p_ilce VARCHAR2, p_il_id NUMBER) AS

BEGIN

INSERT INTO ILCE (ilce, il_id) VALUES (p_ilce, p_il_id);

DBMS_OUTPUT.PUT_LINE('İlçe başarıyla eklendi: ' || p_ilce);

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);

END ekle_ilce;

↑

PROCEDURE ekle_gazete_ad(p_gazete_ad VARCHAR2) AS

BEGIN

INSERT INTO GAZETE_AD (gazete_ad) VALUES (p_gazete_ad);

DBMS_OUTPUT.PUT_LINE('Gazete adı başarıyla eklendi: ' || p_gazete_ad);

EXCEPTION

WHEN OTHERS THEN

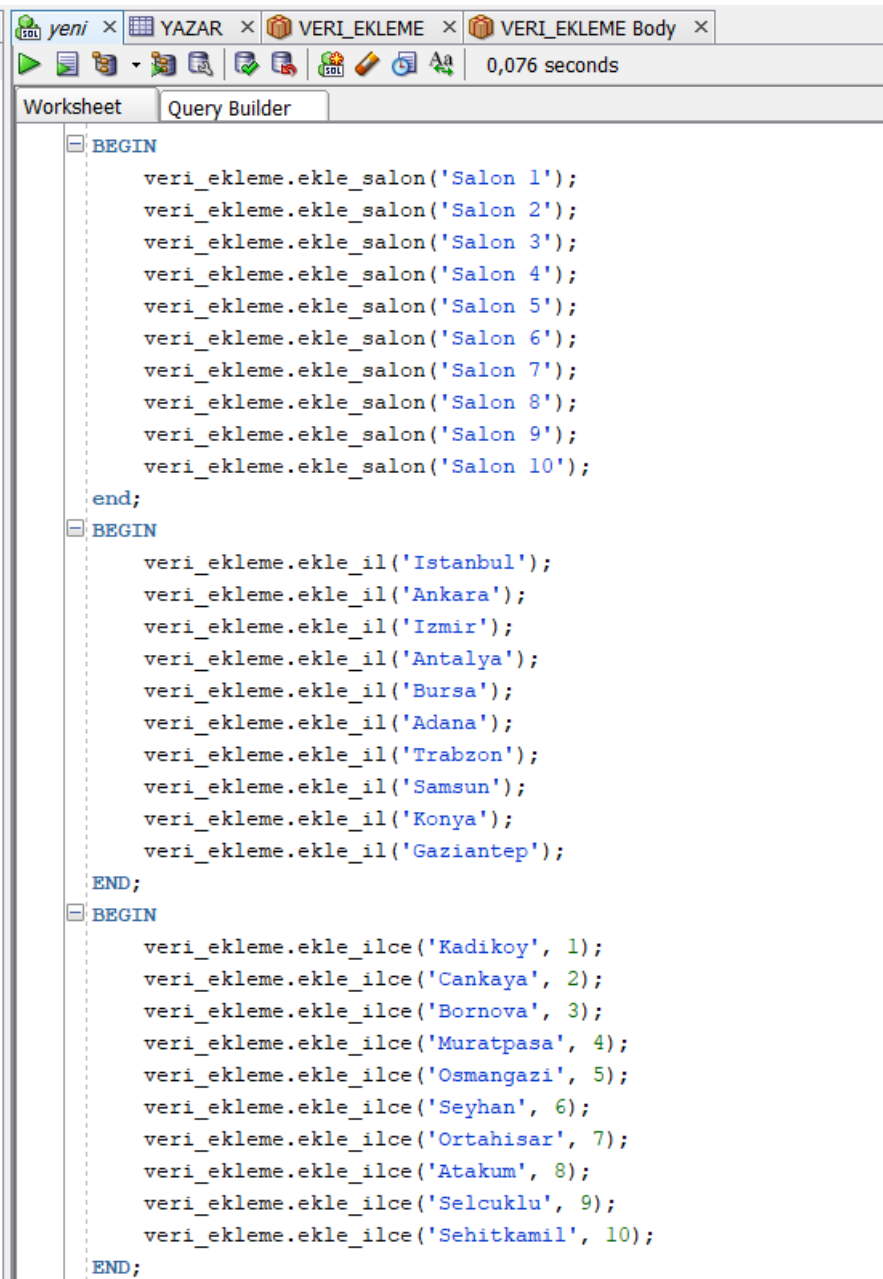
DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);

END ekle_gazete_ad;

PAKET OLUŞTURULDU

Package VERİ_EKLEME compiled

Package Body VERİ_EKLEME compiled



The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for running, saving, and other database operations, with a timer showing 0,076 seconds. The 'Query Builder' tab is active, displaying a PL/SQL script. The script is organized into three blocks, each starting with a 'BEGIN' statement and ending with an 'END;' statement. The first block contains ten calls to the 'veri_ekleme.ekle_salon' procedure, each with a different salon number from 1 to 10. The second block contains ten calls to the 'veri_ekleme.ekle_il' procedure, each with a different city name from 'Istanbul' to 'Gaziantep'. The third block contains ten calls to the 'veri_ekleme.ekle_ilce' procedure, each with a different district name from 'Kadikoy' to 'Sehitkamil' and a numeric value from 1 to 10.

```
BEGIN
    veri_ekleme.ekle_salon('Salon 1');
    veri_ekleme.ekle_salon('Salon 2');
    veri_ekleme.ekle_salon('Salon 3');
    veri_ekleme.ekle_salon('Salon 4');
    veri_ekleme.ekle_salon('Salon 5');
    veri_ekleme.ekle_salon('Salon 6');
    veri_ekleme.ekle_salon('Salon 7');
    veri_ekleme.ekle_salon('Salon 8');
    veri_ekleme.ekle_salon('Salon 9');
    veri_ekleme.ekle_salon('Salon 10');
end;

BEGIN
    veri_ekleme.ekle_il('Istanbul');
    veri_ekleme.ekle_il('Ankara');
    veri_ekleme.ekle_il('Izmir');
    veri_ekleme.ekle_il('Antalya');
    veri_ekleme.ekle_il('Bursa');
    veri_ekleme.ekle_il('Adana');
    veri_ekleme.ekle_il('Trabzon');
    veri_ekleme.ekle_il('Samsun');
    veri_ekleme.ekle_il('Konya');
    veri_ekleme.ekle_il('Gaziantep');
END;

BEGIN
    veri_ekleme.ekle_ilce('Kadikoy', 1);
    veri_ekleme.ekle_ilce('Cankaya', 2);
    veri_ekleme.ekle_ilce('Bornova', 3);
    veri_ekleme.ekle_ilce('Muratpasa', 4);
    veri_ekleme.ekle_ilce('Osmangazi', 5);
    veri_ekleme.ekle_ilce('Seyhan', 6);
    veri_ekleme.ekle_ilce('Ortahisar', 7);
    veri_ekleme.ekle_ilce('Atakum', 8);
    veri_ekleme.ekle_ilce('Selcuklu', 9);
    veri_ekleme.ekle_ilce('Sehitkamil', 10);
END;
```



```
BEGIN
    veri_ekleme.ekle_gazete_ad('Hurriyet');
    veri_ekleme.ekle_gazete_ad('Milliyet');
    veri_ekleme.ekle_gazete_ad('Sabah');
    veri_ekleme.ekle_gazete_ad('Sozcu');
    veri_ekleme.ekle_gazete_ad('Cumhuriyet');
    veri_ekleme.ekle_gazete_ad('Yeni Safak');
    veri_ekleme.ekle_gazete_ad('Vatan');
    veri_ekleme.ekle_gazete_ad('Posta');
    veri_ekleme.ekle_gazete_ad('Zaman');
    veri_ekleme.ekle_gazete_ad('Aksam');

END;

BEGIN
    veri_ekleme.ekle_gazete('1931', 1, 1);
    veri_ekleme.ekle_gazete('1932', 2, 2);
    veri_ekleme.ekle_gazete('1933', 3, 3);
    veri_ekleme.ekle_gazete('1934', 4, 4);
    veri_ekleme.ekle_gazete('1935', 5, 5);
    veri_ekleme.ekle_gazete('1936', 6, 6);
    veri_ekleme.ekle_gazete('1937', 7, 7);
    veri_ekleme.ekle_gazete('1938', 8, 8);
    veri_ekleme.ekle_gazete('1939', 9, 9);
    veri_ekleme.ekle_gazete('1940', 10, 10);
    veri_ekleme.ekle_gazete('1941', 1, 1);
    veri_ekleme.ekle_gazete('1942', 2, 2);
    veri_ekleme.ekle_gazete('1943', 3, 3);
    veri_ekleme.ekle_gazete('1944', 4, 4);
    veri_ekleme.ekle_gazete('1945', 5, 5);
    veri_ekleme.ekle_gazete('1946', 6, 6);
    veri_ekleme.ekle_gazete('1947', 7, 7);
    veri_ekleme.ekle_gazete('1948', 8, 8);
    veri_ekleme.ekle_gazete('1949', 9, 9);
    veri_ekleme.ekle_gazete('1950', 10, 10);
    veri_ekleme.ekle_gazete('1951', 1, 1);
    veri_ekleme.ekle_gazete('1952', 2, 2);
    veri_ekleme.ekle_gazete('1953', 3, 3);
    veri_ekleme.ekle_gazete('1954', 4, 4);
```




```

-- BEGIN
veri_ekleme.ekle_dergi('Dergi 1', 1, 1);
veri_ekleme.ekle_dergi('Dergi 2', 2, 2);
veri_ekleme.ekle_dergi('Dergi 3', 3, 3);
veri_ekleme.ekle_dergi('Dergi 4', 4, 4);
veri_ekleme.ekle_dergi('Dergi 5', 5, 5);
veri_ekleme.ekle_dergi('Dergi 6', 6, 6);
veri_ekleme.ekle_dergi('Dergi 7', 7, 7);
veri_ekleme.ekle_dergi('Dergi 8', 8, 8);
veri_ekleme.ekle_dergi('Dergi 9', 9, 9);
veri_ekleme.ekle_dergi('Dergi 10', 10, 10);
-- END;

-- BEGIN
veri_ekleme.ekle_yazar_kitap(1, 2);
veri_ekleme.ekle_yazar_kitap(1, 1);
veri_ekleme.ekle_yazar_kitap(2, 2);
veri_ekleme.ekle_yazar_kitap(3, 3);
veri_ekleme.ekle_yazar_kitap(4, 4);
veri_ekleme.ekle_yazar_kitap(5, 5);
veri_ekleme.ekle_yazar_kitap(6, 6);
veri_ekleme.ekle_yazar_kitap(7, 7);
veri_ekleme.ekle_yazar_kitap(8, 8);
veri_ekleme.ekle_yazar_kitap(9, 9);
veri_ekleme.ekle_yazar_kitap(10, 10);
-- END;

-- BEGIN
veri_ekleme.ekle_yazar_dergi(1, 1);
veri_ekleme.ekle_yazar_dergi(2, 2);
veri_ekleme.ekle_yazar_dergi(3, 3);
veri_ekleme.ekle_yazar_dergi(4, 4);
veri_ekleme.ekle_yazar_dergi(5, 5);
veri_ekleme.ekle_yazar_dergi(6, 6);
veri_ekleme.ekle_yazar_dergi(7, 7);
veri_ekleme.ekle_yazar_dergi(8, 8);
veri_ekleme.ekle_yazar_dergi(9, 9);
veri_ekleme.ekle_yazar_dergi(10, 10);
-- END;

```

VERİLER EKLENDİ ÜYE KISMINA BİLEREK VERİ EKLEMEDİM ORADA TRİGGER VAR

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

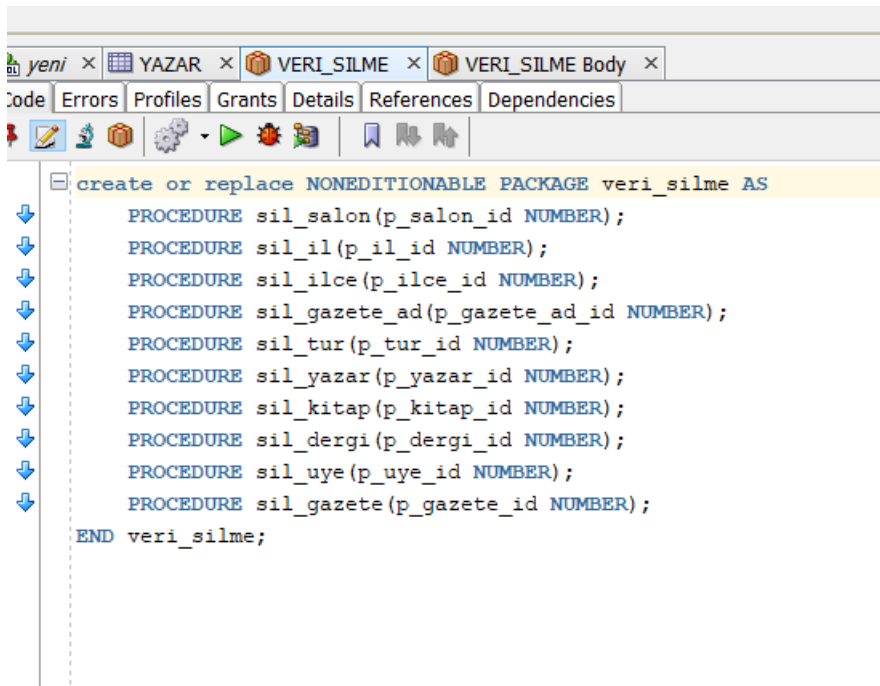
PL/SQL procedure successfully completed.

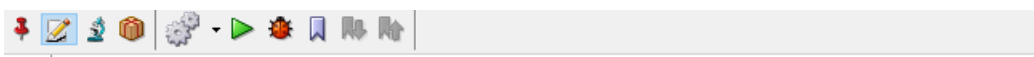
*Action:

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

VERİ SİLME





```
PROCEDURE sil_tur(p_tur_id NUMBER) AS
BEGIN
    DELETE FROM TUR WHERE tur_id = p_tur_id;
    DBMS_OUTPUT.PUT_LINE('Tür başarıyla silindi.');
```

Up arrow icon

```
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);
END sil_tur;
```

Up arrow icon

```
PROCEDURE sil_yazar(p_yazar_id NUMBER) AS
BEGIN
    DELETE FROM YAZAR WHERE yazar_id = p_yazar_id;
    DBMS_OUTPUT.PUT_LINE('Yazar başarıyla silindi.');
```

Up arrow icon

```
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);
END sil_yazar;
```

Up arrow icon

```
PROCEDURE sil_kitap(p_kitap_id NUMBER) AS
BEGIN
    DELETE FROM YAZAR_KITAP WHERE kitap_id = p_kitap_id;
    DBMS_OUTPUT.PUT_LINE('İlişki kaldırıldı.');
```

Up arrow icon

```
DELETE FROM KITAP WHERE kitap_id = p_kitap_id;
    DBMS_OUTPUT.PUT_LINE('Kitap başarıyla silindi.');
```

```
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);
END sil_kitap;
```

Up arrow icon

```
PROCEDURE sil_dergi(p_dergi_id NUMBER) AS
BEGIN
    DELETE FROM YAZAR_DERGI WHERE dergi_id = p_dergi_id;
    DBMS_OUTPUT.PUT_LINE('ilişki kaldırıldı.');
```


Up arrow icon



```
DELETE FROM DERGI WHERE dergi_id = p_dergi_id;
    DBMS_OUTPUT.PUT_LINE('Dergi başarıyla silindi.');
```

```
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);
END sil_dergi;
```



yeni x YAZAR x VERI_SILME x VERI_SILME Body x

CodeErrorsProfilesGrantsDetailsReferencesDependencies





create or replace NONEDITIONABLE PACKAGE BODY veri_silme AS



PROCEDURE sil_salon(p_salon_id NUMBER) AS

BEGIN

DELETE FROM SALON WHERE salon_id = p_salon_id;



DBMS_OUTPUT.PUT_LINE('Salon başarıyla silindi.');

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);

END sil_salon;



PROCEDURE sil_il(p_il_id NUMBER) AS

BEGIN

DELETE FROM IL WHERE il_id = p_il_id;



DBMS_OUTPUT.PUT_LINE('İl başarıyla silindi.');

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);

END sil_il;



PROCEDURE sil_ilce(p_ilce_id NUMBER) AS

BEGIN

DELETE FROM ILCE WHERE ilce_id = p_ilce_id;



DBMS_OUTPUT.PUT_LINE('İlçe başarıyla silindi.');

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);

END sil_ilce;



PROCEDURE sil_gazete_ad(p_gazete_ad_id NUMBER) AS

BEGIN

DELETE FROM GAZETE_AD WHERE gazete_id = p_gazete_ad_id;

DBMS_OUTPUT.PUT_LINE('Gazete adı başarıyla silindi.');

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);

END sil_gazete_ad;

-



```
PROCEDURE sil_uye(p_uye_id NUMBER) AS
BEGIN
    DELETE FROM UYE WHERE uye_id = p_uye_id;
    DBMS_OUTPUT.PUT_LINE('Üye başarıyla silindi.');
```

EXCEPTION

```
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);
END sil_uye;
```



```
PROCEDURE sil_gazete(p_gazete_id NUMBER) AS
BEGIN
    DELETE FROM GAZETE WHERE gazet_id = p_gazete_id;
    DBMS_OUTPUT.PUT_LINE('Gazete başarıyla silindi.');
```

EXCEPTION

```
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);
END sil_gazete;
```

```
END veri_silme;
```

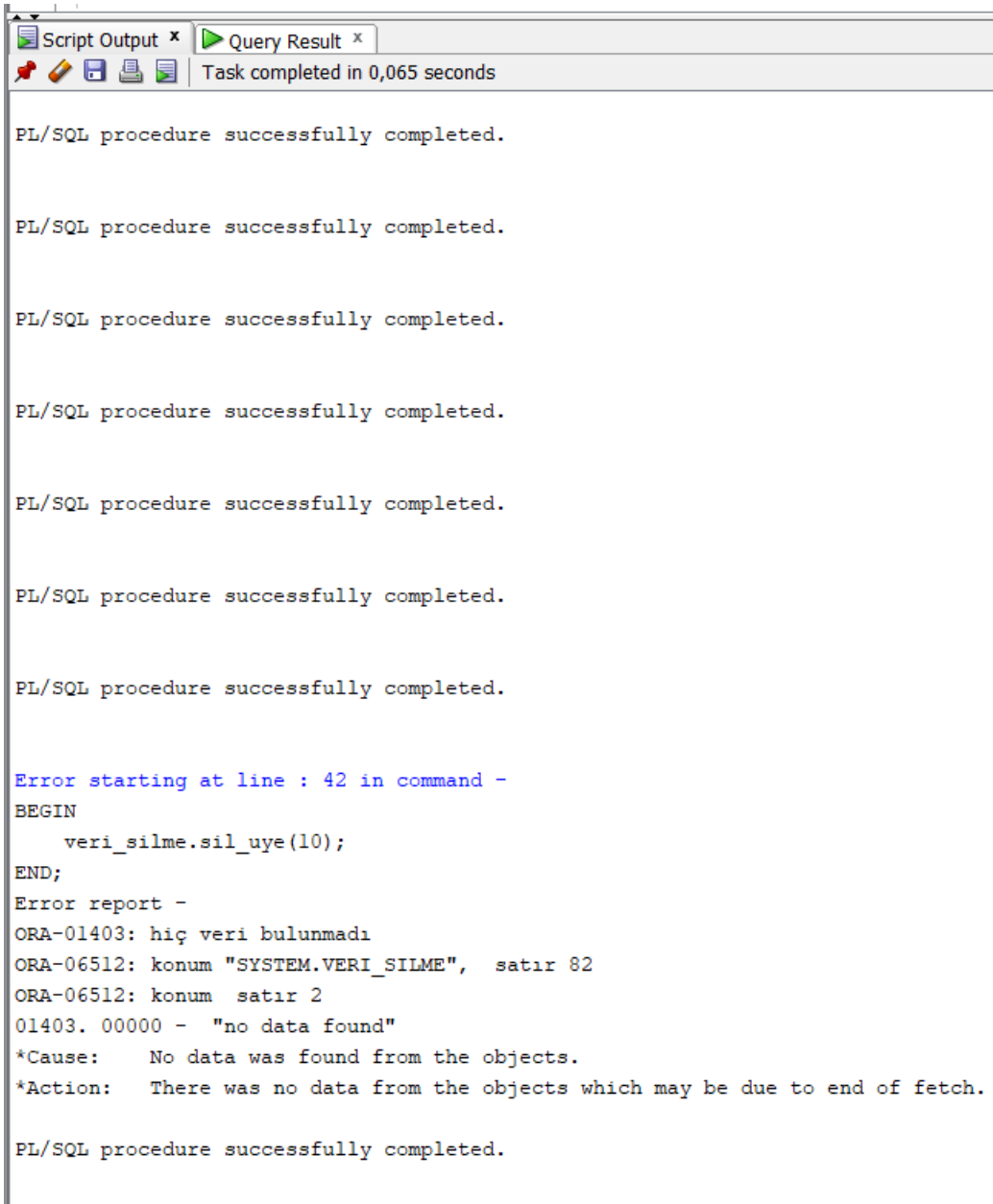
PACKAGE BODY veri_silme > PROCEDURE sil_yazar >

yeni x YAZAR x VERI_SILME x

0,065 seconds

Worksheet Query Builder

```
BEGIN
    veri_silme.sil_salon(10);
END;
BEGIN
    veri_silme.sil_il(10);
END;
BEGIN
    veri_silme.sil_ilce(10);
END;
BEGIN
    veri_silme.sil_gazete_ad(10);
END;
BEGIN
    veri_silme.sil_tur(10);
END;
--BURADA HAREKETTEN DE SİLECEKTİR
BEGIN
    veri_silme.sil_kitap(10);
END;
BEGIN
    veri_silme.sil_dergi(10);
END;
BEGIN
    veri_silme.sil_yazar(10);
END;
BEGIN
    veri_silme.sil_gazete(30);
END;
--UYE TABLOSUNA VERİ EKLEMEDİĞİM İÇİN SİLEMEDİM
--AMA UYE TABLOSUNDAN VERİ SİLEBİLMEM İÇİN KİTAP_ID KISMININ NULL OLMASI GEREKİR
```

Script Output x Query Result x

Task completed in 0,065 seconds

```
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Error starting at line : 42 in command -
BEGIN
    veri_silme.sil_uye(10);
END;
Error report -
ORA-01403: hiç veri bulunmadı
ORA-06512: konum "SYSTEM.VERI_SILME",  satır 82
ORA-06512: konum  satır 2
01403. 00000 - "no data found"
*Cause:      No data was found from the objects.
*Action:     There was no data from the objects which may be due to end of fetch.


PL/SQL procedure successfully completed.
```

Buradaki hata uye kısmından veri silmeye çalıştığım için uyede veri yokki

VERİ GÜNCELLEME

veri yeni x VERI_GUNCELLEME x VERI_GUNCELLEME Body x

Code Errors Profiles Grants Details References Dependencies



create or replace NONEDITIONABLE PACKAGE veri_guncelleme AS

PROCEDURE guncelle_salon_adı(p_salon_id IN NUMBER, p_yeni_ad IN VARCHAR2);

PROCEDURE guncelle_kitap_sayısı(p_kitap_id IN NUMBER, p_yeni_sayı IN NUMBER);

PROCEDURE guncelle_kitap_salonu(p_kitap_id IN NUMBER, p_yeni_salon IN NUMBER);

PROCEDURE guncelle_dergi_salonu(p_dergi_id IN NUMBER, p_yeni_salon IN NUMBER);

PROCEDURE guncelle_uye_kitap(p_uye_id IN NUMBER, p_kitap_id IN NUMBER);


PROCEDURE guncelle_uye_adres(p_uye_id IN NUMBER, p_il_id IN NUMBER, p_ilce_id IN NUMBER);

PROCEDURE guncelle_uye_telefon(p_uye_id IN NUMBER, p_yeni_telefon IN VARCHAR2);

END veri_guncelleme;

veri yeni x VERI_GUNCELLEME x VERI_GUNCELLEME Body x

Code Errors Profiles Grants Details References Dependencies



create or replace NONEDITIONABLE PACKAGE BODY veri_guncelleme AS

PROCEDURE guncelle_salon_adı(p_salon_id IN NUMBER, p_yeni_ad IN VARCHAR2) IS

BEGIN

UPDATE SALON

SET salonlar = p_yeni_ad

WHERE salon_id = p_salon_id;

END guncelle_salon_adı;

PROCEDURE guncelle_kitap_sayısı(p_kitap_id IN NUMBER, p_yeni_sayı IN NUMBER) IS

BEGIN

UPDATE KITAP

SET kitap_sayısı = p_yeni_sayı

WHERE kitap_id = p_kitap_id;

END guncelle_kitap_sayısı;

PROCEDURE guncelle_kitap_salonu(p_kitap_id IN NUMBER, p_yeni_salon IN NUMBER) IS

BEGIN

UPDATE KITAP

SET salon_id = p_yeni_salon

WHERE kitap_id = p_kitap_id;

END guncelle_kitap_salonu;

PROCEDURE guncelle_dergi_salonu(p_dergi_id IN NUMBER, p_yeni_salon IN NUMBER) IS

BEGIN

UPDATE DERGI

SET salon_id = p_yeni_salon

WHERE dergi_id = p_dergi_id;

END guncelle_dergi_salonu;

PROCEDURE guncelle_uye_kitap(p_uye_id IN NUMBER, p_kitap_id IN NUMBER) IS

BEGIN

UPDATE UYE

SET kitap_id = p_kitap_id

WHERE uye_id = p_uye_id;

END guncelle_uye_kitap;

```
END guncelle_uye_kitap;
```

```
PROCEDURE guncelle_uye_adres(p_uye_id IN NUMBER, p_il_id IN NUMBER, p_ilce_id IN NUMBER) IS  
BEGIN
```

```
    UPDATE UYE
```

```
    SET il_id = p_il_id,
```

```
        ilce_id = p_ilce_id
```

```
    WHERE uye_id = p_uye_id;
```

```
END guncelle_uye_adres;
```

```
PROCEDURE guncelle_uye_telefon(p_uye_id IN NUMBER, p_yeni_telefon IN VARCHAR2) IS  
BEGIN
```

```
    UPDATE UYE
```

```
    SET uye_tel = p_yeni_telefon
```

```
    WHERE uye_id = p_uye_id;
```

```
END guncelle_uye_telefon;
```

```
END veri_guncelleme;
```

PACKAGE BODY veri_guncelleme > PROCEDURE guncelle_kitap_salonu

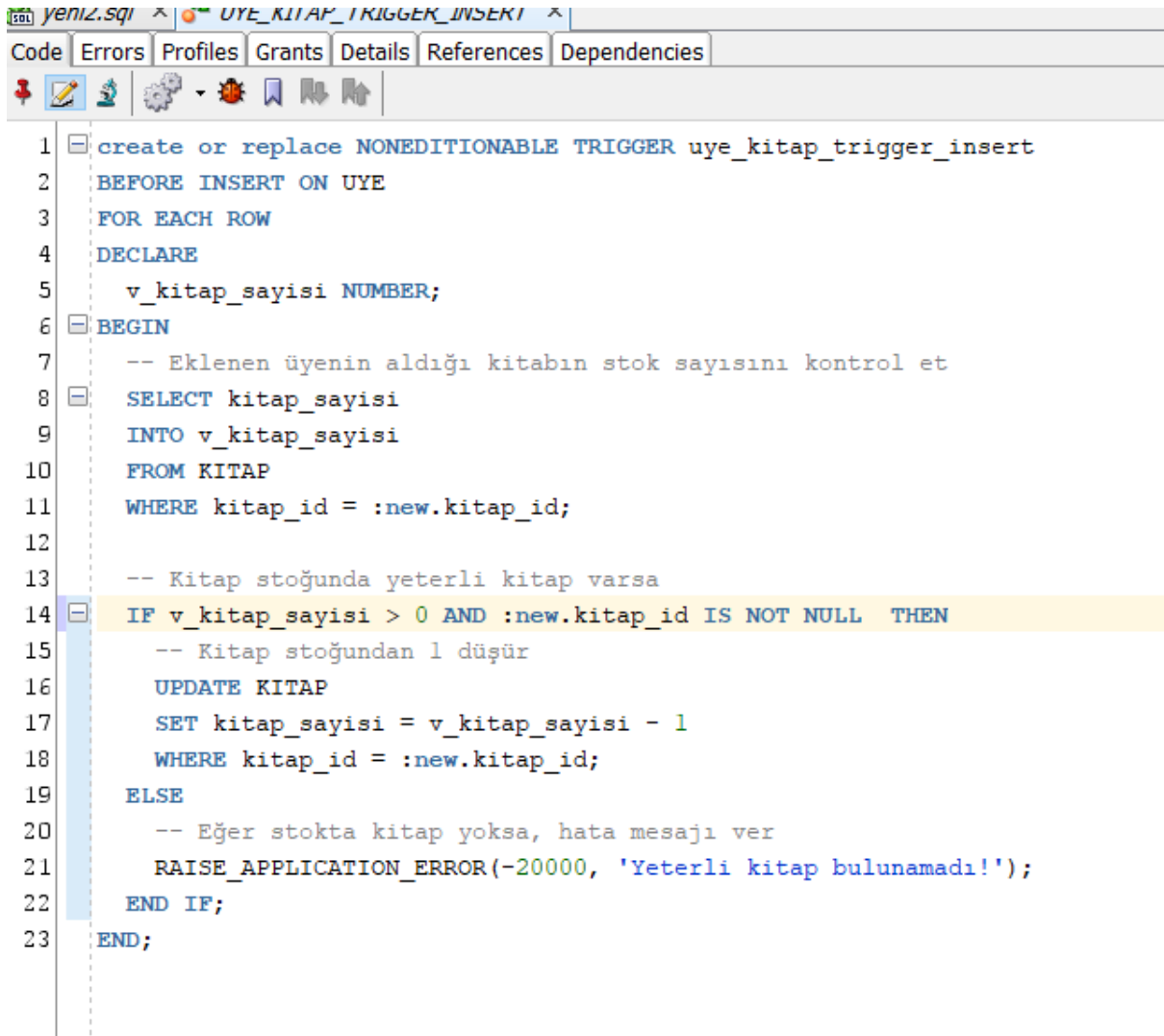
<

1 BEGIN
2 veri_guncelleme.guncelle_kitap_salonu(7, 2);
3 veri_guncelleme.guncelle_dergi_salonu(9, 4);
4 veri_guncelleme.guncelle_uye_adres(13, 3, 5);
5 veri_guncelleme.guncelle_uye_telefon(15, '555-1234');
6 END;
7
8 BEGIN
9 veri_guncelleme.guncelle_kitap_sayısı(1, 10);
10 veri_guncelleme.guncelle_kitap_sayısı(2, 10);
11 veri_guncelleme.guncelle_kitap_sayısı(3, 10);
12 veri_guncelleme.guncelle_kitap_sayısı(4, 10);
13 veri_guncelleme.guncelle_kitap_sayısı(5, 10);
14 veri_guncelleme.guncelle_kitap_sayısı(6, 10);
15 veri_guncelleme.guncelle_kitap_sayısı(7, 10);
16 veri_guncelleme.guncelle_kitap_sayısı(8, 10);
17 veri_guncelleme.guncelle_kitap_sayısı(9, 10);
18 END;
19
20

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

TRİGGERLAR UYE EKLENİNCE ALDIĞI KİTAP SONUCUNDA KİTAP STOĞUNUN DÜŞMESİ



The screenshot shows an SQL IDE window with a tab titled 'UYE_KITAP_TRIGGER_INSERT'. The code editor displays a PL/SQL trigger named 'uye_kitap_trigger_insert' that is triggered before an insert on the 'UYE' table. The trigger declares a variable 'v_kitap_sayisi' of type 'NUMBER'. It then performs a 'SELECT' query to retrieve the 'kitap_sayisi' from the 'KITAP' table where 'kitap_id' matches the new record's 'kitap_id'. An 'IF' statement checks if 'v_kitap_sayisi' is greater than 0 and if ':new.kitap_id' is not null. If both conditions are true, it updates the 'KITAP' table by decrementing 'kitap_sayisi' by 1. If the conditions are not met, it raises an application error with the message 'Yeterli kitap bulunamadı!'. The trigger ends with 'END;'.

```
1 create or replace NONEDITIONABLE TRIGGER uye_kitap_trigger_insert
2 BEFORE INSERT ON UYE
3 FOR EACH ROW
4 DECLARE
5     v_kitap_sayisi NUMBER;
6 BEGIN
7     -- Eklenen üyenin aldığı kitabın stok sayısını kontrol et
8     SELECT kitap_sayisi
9     INTO v_kitap_sayisi
10    FROM KITAP
11   WHERE kitap_id = :new.kitap_id;
12
13     -- Kitap stoğunda yeterli kitap varsa
14     IF v_kitap_sayisi > 0 AND :new.kitap_id IS NOT NULL THEN
15         -- Kitap stoğundan 1 düşür
16         UPDATE KITAP
17         SET kitap_sayisi = v_kitap_sayisi - 1
18         WHERE kitap_id = :new.kitap_id;
19     ELSE
20         -- Eğer stokta kitap yoksa, hata mesajı ver
21         RAISE_APPLICATION_ERROR(-20000, 'Yeterli kitap bulunamadı!');
22     END IF;
23 END;
```

ŞİMDİ UYE KISMINA EKLEMELER YAPIYORUZ

ÖNCESİ

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

```
select*from UYE;  
  
BEGIN veri_ekleme.ekle_uye('Sema', 'Kurt', 1, '12345678901', '5551234567', 1, 1); END;  
BEGIN veri_ekleme.ekle_uye('Cem', 'Aslan', 2, '23456789012', '5552345678', 2, 2); END;  
BEGIN veri_ekleme.ekle_uye('Deniz', 'Turan', 3, '34567890123', '5553456789', 3, 3); END;  
BEGIN veri_ekleme.ekle_uye('Ahmet', 'Kaya', 4, '45678901234', '5554567890', 4, 4); END;  
BEGIN veri_ekleme.ekle_uye('Mehmet', 'Yildiz', 5, '56789012345', '5555678901', 5, 5); END;  
BEGIN veri_ekleme.ekle_uye('Ayşe', 'Kara', 6, '67890123456', '5556789012', 6, 6); END;  
BEGIN veri_ekleme.ekle_uye('Fatma', 'Yılmaz', 7, '78901234567', '5557890123', 7, 7); END;  
BEGIN veri_ekleme.ekle_uye('Mustafa', 'Demir', 8, '89012345678', '5558901234', 8, 8); END;  
BEGIN veri_ekleme.ekle_uye('Emine', 'Çelik', 9, '90123456789', '5559012345', 9, 9); END;  
  
select*from KITAP;  
select*from UYE;
```

Script Output x Query Result x Query Result 1 x

SQL All Rows Fetched: 0 in 0,002 seconds

UYE_ID	UYE_AD	UYE_S...	KITAP_ID	UYE_TC	UYE_TEL	IL_ID	ILCE_ID
--------	--------	----------	----------	--------	---------	-------	---------

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

```
select*from KITAP;  
select*from UYE;  
  
BEGIN veri_ekleme.ekle_uye('Sema', 'Kurt', 1, '12345678901', '5551234567', 1, 1); END;  
BEGIN veri_ekleme.ekle_uye('Cem', 'Aslan', 2, '23456789012', '5552345678', 2, 2); END;  
BEGIN veri_ekleme.ekle_uye('Deniz', 'Turan', 3, '34567890123', '5553456789', 3, 3); END;  
BEGIN veri_ekleme.ekle_uye('Ahmet', 'Kaya', 4, '45678901234', '5554567890', 4, 4); END;  
BEGIN veri_ekleme.ekle_uye('Mehmet', 'Yildiz', 5, '56789012345', '5555678901', 5, 5); END;  
BEGIN veri_ekleme.ekle_uye('Ayşe', 'Kara', 6, '67890123456', '5556789012', 6, 6); END;  
BEGIN veri_ekleme.ekle_uye('Fatma', 'Yılmaz', 7, '78901234567', '5557890123', 7, 7); END;  
BEGIN veri_ekleme.ekle_uye('Mustafa', 'Demir', 8, '89012345678', '5558901234', 8, 8); END;  
BEGIN veri_ekleme.ekle_uye('Emine', 'Çelik', 9, '90123456789', '5559012345', 9, 9); END;  
  
select*from KITAP;  
select*from UYE;
```

Script Output x Query Result x Query Result 1 x

SQL All Rows Fetched: 9 in 0,003 seconds

KITAP_ID	KITAP_AD	KITAP_SAYISI	SALON_ID	TUR_ID
1	1 Kitap 1	10	1	1
2	2 Kitap 2	10	2	2
3	3 Kitap 3	10	3	3
4	4 Kitap 4	10	4	4
5	5 Kitap 5	10	5	5
6	6 Kitap 6	10	6	6
7	7 Kitap 7	10	2	7
8	8 Kitap 8	10	8	8
9	9 Kitap 9	10	9	9

SONRASI

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

SQL All Rows Fetched: 9 in 0,003 seconds

UYE_ID	UYE_AD	UYE_SOYAD	KITAP_ID	UYE_TC	UYE_TEL	IL_ID	ILCE_ID
1	1 Sema	Kurt		1 12345678901	5551234567	1	1
2	2 Cem	Aslan		2 23456789012	5552345678	2	2
3	3 Deniz	Turan		3 34567890123	5553456789	3	3
4	4 Ahmet	Kaya		4 45678901234	5554567890	4	4
5	5 Mehmet	Yildiz		5 56789012345	5555678901	5	5
6	6 Ayşe	Kara		6 67890123456	5556789012	6	6
7	7 Fatma	Yılmaz		7 78901234567	5557890123	7	7
8	8 Mustafa	Demir		8 89012345678	5558901234	8	8
9	9 Emine	Çelik		9 90123456789	5559012345	9	9

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

SQL All Rows Fetched: 9 in 0,003 seconds


	KITAP_ID	KITAP_AD	KITAP_SAYISI	SALON_ID	TUR_ID
1	1	Kitap 1		9	1
2	2	Kitap 2		9	2
3	3	Kitap 3		9	3
4	4	Kitap 4		9	4
5	5	Kitap 5		9	5
6	6	Kitap 6		9	6
7	7	Kitap 7		9	2
8	8	Kitap 8		9	8
9	9	Kitap 9		9	9

Diğer trigger update edilince diyelimki üye 9 9.kitabı bırakıp 8 i aldı

yada üye 9 direkt kitabı bıraktı iki durum için de deneyelim

yeni x UYE_KITAP_TRIGGER_UPDATE x

CodeDependenciesErrorsProfilesGrantsReferencesDetails



```
1 create or replace NONEDITIONABLE TRIGGER uye_kitap_trigger_UPDATE
2 AFTER UPDATE ON UYE
3 FOR EACH ROW
4 DECLARE
5     v_eski_kitap_id NUMBER;
6     v_yeni_kitap_id NUMBER;
7 BEGIN
8     SELECT :old.kitap_id INTO v_eski_kitap_id FROM DUAL;
9
10    SELECT :new.kitap_id INTO v_yeni_kitap_id FROM DUAL;
11
12    IF v_eski_kitap_id IS NOT NULL AND v_yeni_kitap_id IS NULL THEN
13
14        UPDATE KITAP
15        SET kitap_sayisi = kitap_sayisi + 1
16        WHERE kitap_id = v_eski_kitap_id;
17    ELSE
18        IF v_eski_kitap_id IS NULL and v_yeni_kitap_id IS NOT NULL THEN
19
20            UPDATE KITAP
21            SET kitap_sayisi = kitap_sayisi - 1
22            WHERE kitap_id = v_yeni_kitap_id;
23        ELSE
24
25            IF v_eski_kitap_id <> v_yeni_kitap_id THEN
26
27                UPDATE KITAP
28                SET kitap_sayisi = kitap_sayisi + 1
29                WHERE kitap_id = v_eski_kitap_id;
30
31                UPDATE KITAP
32                SET kitap_sayisi = kitap_sayisi - 1
33                WHERE kitap_id = v_yeni_kitap_id;
34            END IF;
35        END IF;
36    END IF;
37 END;
```

İLK HAL

Query Result 1 x All Rows Fetched: 9 in 0,003 seconds								Query Result 2 x All Rows Fetched: 9 in 0,002 seconds				
UYE_ID	UYE_AD	UYE_SOYAD	KITAP_ID	UYE_TC	UYE_TEL	IL_ID	ILCE_ID	KITAP_ID	KITAP_AD	KITAP_SAYISI	SALON_ID	TUR_ID
1	Sema	Kurt	1	12345678901	5551234567	1	1	1	Kitap 1	9	1	1
2	Cem	Aslan	2	23456789012	5552345678	2	2	2	Kitap 2	9	2	2
3	Deniz	Turan	3	34567890123	5553456789	3	3	3	Kitap 3	9	3	3
4	Ahmet	Kaya	4	45678901234	5554567890	4	4	4	Kitap 4	9	4	4
5	Mehmet	Yıldız	5	56789012345	5555678901	5	5	5	Kitap 5	9	5	5
6	Ayşe	Kara	6	67890123456	5556789012	6	6	6	Kitap 6	9	6	6
7	Fatma	Yılmaz	7	78901234567	5557890123	7	7	7	Kitap 7	9	2	7
8	Mustafa	Demir	8	89012345678	5558901234	8	8	8	Kitap 8	9	8	8
9	Emine	Çelik	9	90123456789	5559012345	9	9	9	Kitap 9	9	9	9

TRİGGER ÇALIŞTIKTAN SONRA

1	
2	-- 9 İD YE SAHİP ÜYE 8. KİTABI ALACAK 9 U BIRAKMIŞ OLACAK
3	SELECT*FROM UYE;
4	SELECT*FROM KITAP;
5	
6	BEGIN
7	veri_guncelleme.guncelle_uye_kitap(9,8);
8	END;
9	SELECT*FROM UYE;
0	SELECT*FROM KITAP;

Query Result 1 x All Rows Fetched: 9 in 0,003 seconds								Query Result 2 x All Rows Fetched: 9 in 0,003 seconds				
UYE_ID	UYE_AD	UYE_SOYAD	KITAP_ID	UYE_TC	UYE_TEL	IL_ID	ILCE_ID	UYE_ID	UYE_AD	UYE_SOYAD	KITAP_ID	UYE_TC
1	Sema	Kurt	1	12345678901	5551234567	1	1	1	Sema	Kurt	1	12345678901
2	Cem	Aslan	2	23456789012	5552345678	2	2	2	Cem	Aslan	2	23456789012
3	Deniz	Turan	3	34567890123	5553456789	3	3	3	Deniz	Turan	3	34567890123
4	Ahmet	Kaya	4	45678901234	5554567890	4	4	4	Ahmet	Kaya	4	45678901234
5	Mehmet	Yıldız	5	56789012345	5555678901	5	5	5	Mehmet	Yıldız	5	56789012345
6	Ayşe	Kara	6	67890123456	5556789012	6	6	6	Ayşe	Kara	6	67890123456
7	Fatma	Yılmaz	7	78901234567	5557890123	7	7	7	Fatma	Yılmaz	7	78901234567
8	Mustafa	Demir	8	89012345678	5558901234	8	8	8	Mustafa	Demir	8	89012345678
9	Emine	Çelik	8	90123456789	5559012345	9	9	9	Emine	Çelik	8	90123456789

KITAP_ID	KITAP_AD	KITAP_SAYISI	SALON_ID	TUR_ID
1	1 Kitap 1	9	1	1
2	2 Kitap 2	9	2	2
3	3 Kitap 3	9	3	3
4	4 Kitap 4	9	4	4
5	5 Kitap 5	9	5	5
6	6 Kitap 6	9	6	6
7	7 Kitap 7	9	2	7
8	8 Kitap 8	8	8	8
9	9 Kitap 9	10	9	9

ŞİMDİDE 9 İD YE SAHİP OLAN ÜYE KİTABI BIRAKSIN

```

1
2  -- 9 İD YE SAHİP ÜYE 8. KİTABI ALACAK 9 U BIRAKMIŞ OLACAK
3  SELECT*FROM UYE;
4  SELECT*FROM KİTAP;
5
6  BEGIN
7      veri_guncelleme.guncelle_uye_kitap(9,NULL);
8  END;
9  SELECT*FROM UYE;
10 SELECT*FROM KİTAP;

```

KITAP_ID	KITAP_AD	KITAP_SAYISI	SALON_ID	TUR_ID
1	1 Kitap 1	9	1	1
2	2 Kitap 2	9	2	2
3	3 Kitap 3	9	3	3
4	4 Kitap 4	9	4	4
5	5 Kitap 5	9	5	5
6	6 Kitap 6	9	6	6
7	7 Kitap 7	9	2	7
8	8 Kitap 8	9	8	8
9	9 Kitap 9	10	9	9

Query Result x Query Result 1 x Script Output x Query Result 2 x Query Result 3 x Query Result 4 x Query Res								
All Rows Fetched: 9 in 0,002 seconds								
	UYE_ID	UYE_AD	UYE_SOYAD	KITAP_ID	UYE_TC	UYE_TEL	IL_ID	ILCE_ID
1	1	Sema	Kurt	1	12345678901	5551234567	1	1
2	2	Cem	Aslan	2	23456789012	5552345678	2	2
3	3	Deniz	Turan	3	34567890123	5553456789	3	3
4	4	Ahmet	Kaya	4	45678901234	5554567890	4	4
5	5	Mehmet	Yildiz	5	56789012345	5555678901	5	5
6	6	Ayşe	Kara	6	67890123456	5556789012	6	6
7	7	Fatma	Yılmaz	7	78901234567	5557890123	7	7
8	8	Mustafa	Demir	8	89012345678	5558901234	8	8
9	9	Emine	Çelik	(null)	90123456789	5559012345	9	9

tekrar eden kayıtlar silme kodu basarili
çalışmad

CREATE OR REPLACE PACKAGE BODY tekrar AS

PROCEDURE sil_tekrar_eden_kayıtlar (

p_tablo_adı IN VARCHAR2,

p_tanımlayıcı_sütun IN VARCHAR2

) AS

BEGIN

-- Dinamik SQL kullanarak tablo adını tanımla

FOR rec IN (

SELECT p_tanımlayıcı_sütun

FROM (SELECT * FROM p_tablo_adı) -- Dinamik tablo adı

GROUP BY p_tanımlayıcı_sütun

HAVING COUNT(*) > 1

) LOOP

DELETE FROM p_tablo_adı

WHERE ROWID NOT IN (

SELECT MIN(ROWID)

FROM p_tablo_adı

WHERE p_tanımlayıcı_sütun = rec.p_tanımlayıcı_sütun

GROUP BY p_tanımlayıcı_sütun

);

```
END LOOP;
```

```
DBMS_OUTPUT.PUT_LINE('Tekrar eden kayıtlar silindi.');
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);
```

```
END sil_tekrar_eden_kayitlar;
```

```
END tekrar;
```

```
CREATE OR REPLACE NONEDITIONABLE PACKAGE BODY tekrar AS
```

```
PROCEDURE sil_tekrar_eden_kayitlar (
```

```
p_tablo_adı IN VARCHAR2,
```

```
p_tanımlayıcı_sütun IN VARCHAR2
```

```
) AS
```

```
v_sql VARCHAR2(1000);
```

```
BEGIN
```

```
-- Dinamik SQL kullanarak tablo adını tanımla
```

```
v_sql := 'DELETE FROM ' || p_tablo_adı ||
```

```
' WHERE ROWID NOT IN (SELECT MIN(ROWID) FROM ' || p_tablo_adı ||
```

```
' GROUP BY ' || p_tanımlayıcı_sütun || ' HAVING COUNT(*) > 1)';
```

```
DBMS_OUTPUT.PUT_LINE('SQL ifadesi: ' || v_sql); -- SQL ifadesini kontrol et
```

```
EXECUTE IMMEDIATE v_sql;
```

```
DBMS_OUTPUT.PUT_LINE('Tekrar eden kayıtlar silindi.');
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
DBMS_OUTPUT.PUT_LINE('Bir hata oluştu: ' || SQLERRM);
```

END sil_tekrar_eden_kayitlar;

END tekrar;