



**FATİH
SULTAN
MEHMET**
VAKIF ÜNİVERSİTESİ

ADI-SOYADI	HALİS FURKAN AKGÜL
OKUL NUMARASI	1821221018
BAŞLIK	VERİTABANI BÜTÜNLEME ÖDEVİ
KONUSU	KİŞİSEL NOT TAKİP SİSTEMİ

ANALİZ BELGESİ:

Proje konum Kişisel Not Takip Sistemi'dir. Veri Modelim 20 tablodan oluşmaktadır.

1)Ders: Tablomda DERS_KODU, DERS_ADI, AKTS, DERSIN_OGRETMENI_ID olmak üzere 4 adet kolon bulunmakta, bu kolonlara göre ders verisi tutulmaktadır. Bu veriler sayesinde dersin koduna ve o dersi hangi öğretmenin anlattığı bilgisine ulaşabilirsiniz. Bir dersi birden fazla öğrenci alabilir. Bir e çok ilişkileri vardır.

2)Sınav Sonuçları: Tablomda ID, OGRENCI_ID, DERS_ID, NOT, ILAN_TARIHI olmak üzere 5 adet kolon bulunmakta, bu kolonlara göre sınav sonuçlarının verisi tutulmaktadır. Bu veriler sayesinde hangi öğrencinin hangi dersten ne kadar puan aldığı bilgisine ulaşabilirsiniz. Öğrenci tablosundan faydalanılarak OGRENCI_ID kolonu ile eşleyip öğrencinin diğer bilgilerine de ulaşabileceksiniz.

3)Öğrenci: Tablomda OGRENCI_NUMARASI, AD, SOYAD, MAIL, BOLUM_ID, KACINCI_SINIF olmak üzere 6 adet kolon bulunmakta, bu kolonlara göre öğrencilerin verisi tutulmaktadır. Bu veriler sayesinde öğrencilerin okul numaralarına, hangi bölümde kaçınıcı sınıfta olduklarına dair bilgiye ulaşabilirsiniz. Bölüm tablosundan faydalanılarak BOLUM_ID kolonu ile eşleyip bölümlerin diğer bilgilerine de ulaşabileceksiniz.

4)Öğretmen: Tablomda OGRETMEN_ID, AD, SOYAD, MAIL, STATÜ olmak üzere 5 adet kolon bulunmakta, bu kolonlara göre öğretmenlerin verisi tutulmaktadır. Bu veriler sayesinde öğretmenlerin mail adreslerine ve statü olarak hangi konumda olduğunun bilgisine ulaşabilirsiniz.

5)Transkript: Tablomda ID, OGRENCI_ID, DERS_ID, HARF_NOTU olmak üzere 4 adet kolon bulunmakta, bu kolonlara göre öğrencilerin transkript verisi tutulmaktadır. Bu veriler sayesinde öğrencilerin hangi derslerden hangi harf notlarını aldığına dair bilgilere ulaşabilirsiniz. Öğrenci ve DERS tablolarından faydalanılarak OGRENCI_ID, DERS_ID kolonları ile eşleyip transkript alanında öğrencinin ve hangi derse ait ise dersin diğer bilgilerine de ulaşabileceksiniz.

6)Sınav Takvimi: Tablomda ID, OGRENCI_ID, DERS_ID, SINAV_TARIHI, SINAV_YERI olmak üzere 5 adet kolon bulunmakta, bu kolonlara göre sınavların verisi tutulmaktadır. Bu veriler sayesinde öğrencinin ne zaman hangi sınava gireceği hakkında bilgilere ulaşabilirsiniz. Öğrenci ve DERS tablolarından faydalanılarak OGRENCI_ID, DERS_ID kolonları ile eşleyip sınavlar hakkında öğrencinin ve hangi derse ait ise dersin diğer bilgilerine de ulaşabileceksiniz.

7)Okul: Tablomda ID, AD, ADRES olmak üzere 3 adet kolon bulunmakta, bu kolonlara göre okulların verisi tutulmaktadır. Bu veriler sayesinde okulların adları ve adresleri hakkında bilgilere ulaşabilirsiniz.

8)Ders Programı: Tablomda ID, OGRENCI_ID, DERS_GUNU, DERS_ID olmak üzere 4 adet kolon bulunmakta, bu kolonlara göre ders programlarının verisi tutulmaktadır. Bu veriler sayesinde öğrencilerin hangi gün hangi derse gireceği hakkında bilgilere ulaşabilirsiniz. Öğrenci ve DERS tablolarından faydalanılarak OGRENCI_ID, DERS_ID kolonları ile eşleyip ders programı hakkında öğrencilerin ve derslerin diğer bilgilerine de ulaşabileceksiniz.

9)Ödevler: Tablomda ID, OGRENCI_ID, DERS_ID, SON_TESLIM_TARIHI olmak üzere 4 adet kolon bulunmakta, bu kolonlara göre ödevlerin verisi tutulmaktadır. Bu veriler sayesinde hangi öğrencinin hangi dersten ödevi olduğu ve bu ödevin son teslim tarihi hakkında bilgilere ulaşabilirsiniz. Öğrenci ve DERS tablolarından faydalanılarak OGRENCI_ID, DERS_ID kolonları ile eşleyip ödevler hakkında öğrencilerin ve derslerin diğer bilgilerine de ulaşabileceksiniz.

10)Hes Kodu: Tablomda ID, OGRENCI_ID, HES_KODU, DURUM olmak üzere 4 adet kolon bulunmakta, bu kolonlara göre hes kodlarının verileri tutulmaktadır. Bu veriler sayesinde herhangi bir öğrencinin hes kodu ne, durumu ne bu soruların cevaplarına ulaşabilirsiniz. Öğrenci tablosundan faydalanılarak OGRENCI_ID kolonu ile eşleyip hes kodları hakkında öğrencilerin diğer bilgilerine de ulaşabileceksiniz.

11)Duyurular: Tablomda ID, DUYURU olmak üzere 2 adet kolon bulunmakta, bu kolonlara göre duyuruların verileri tutulmaktadır. Bu veriler sayesinde okuldaki duyurular hakkında bilgilere ulaşabilirsiniz.

12)İlişik Kesme: Tablomda ID, OGRENCI_ID, DURUM olmak üzere 3 adet kolon bulunmakta, bu kolonlara göre ilişki durumlarının verileri tutulmaktadır. Bu veriler sayesinde öğrencilerin okul ile olan ilişki durumları hakkında bilgilere ulaşabilirsiniz. Öğrenci tablosundan faydalanılarak OGRENCI_ID kolonu ile eşleyip ilişki kesme hakkında öğrencilerin diğer bilgilerine de ulaşabileceksiniz.

13)Stajlar: Tablomda ID, OGRENCI_ID, STAJ_YERI, STAJ_TURU, DURUM olmak üzere 5 adet kolon bulunmakta, bu kolonlara göre stajların verileri tutulmaktadır. Bu veriler sayesinde öğrencilerin staj konusunda nerede staj yaptıkları ve bu stajlarını gönüllü mü yoksa zorunlu mu yaptıkları hakkında bilgilere ulaşabilirsiniz. Öğrenci tablosundan faydalanılarak OGRENCI_ID kolonu ile eşleyip stajlar hakkında öğrencilerin diğer bilgilerine de ulaşabileceksiniz.

14)Telefon Numaraları: Tablomda ID, OGRENCI_ID, TELEFON_NO olmak üzere 3 adet kolon bulunmakta, bu kolonlara göre telefon numaraları verileri tutulmaktadır. Bu veriler sayesinde öğrencilerin telefon numaraları hakkında bilgilere ulaşabilirsiniz. Bir e çok ilişkileri vardır. Öğrenci tablosundan faydalanılarak OGRENCI_ID kolonu ile eşleyip telefon numaraları hakkında öğrencilerin diğer bilgilerine de ulaşabileceksiniz.

15)Mailler: Tablomda ID, OGRENCI_ID, MAIL_ADRESI olmak üzere 3 adet kolon bulunmakta, bu kolonlara göre mail verileri tutulmaktadır. Bu veriler sayesinde öğrencilerin mailleri hakkında bilgilere ulaşabilirsiniz. Bir e çok ilişkileri vardır. Öğrenci tablosundan faydalanılarak OGRENCI_ID kolonu ile eşleyip öğrencilerin diğer bilgilerine de ulaşabileceksiniz.

16)Adres: Tablomda ID, OGRENCI_ID, ADRES olmak üzere 3 adet kolon bulunmakta, bu kolonlara göre adres verileri tutulmaktadır. Bu veriler sayesinde öğrencilerin adresleri hakkında bilgilere ulaşabilirsiniz. Bir e çok ilişkileri vardır. Öğrenci tablosundan faydalanılarak OGRENCI_ID kolonu ile eşleyip öğrencilerin diğer bilgilerine de ulaşabileceksiniz.

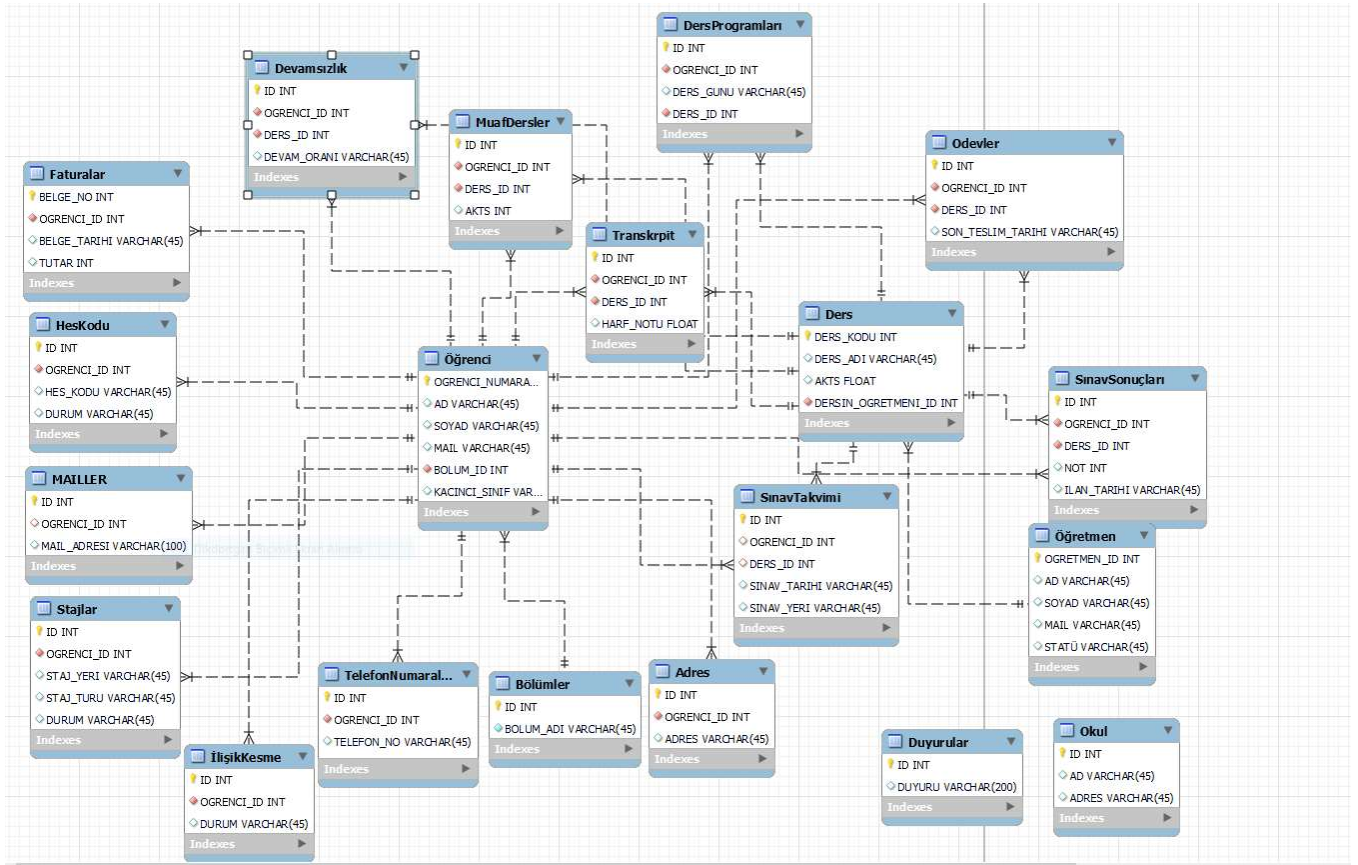
17)Muaf Dersler: Tablomda ID, OGRENCI_ID, DERS_ID, AKTS olmak üzere 4 adet kolon bulunmakta, bu kolonlara göre muaf derslerin verileri tutulmaktadır. Bu veriler sayesinde öğrencilerin muaf olduğu dersler hakkında bilgilere ulaşabilirsiniz. Bir e çok ilişkileri vardır. Öğrenci ve DERS tablolarından faydalanılarak OGRENCI_ID, DERS_ID kolonları ile eşleyip öğrencilerin ve derslerin diğer bilgilerine de ulaşabileceksiniz.

18)Devamsızlık: Tablomda ID, OGRENCI_ID, DERS_ID, DEVAM_ORANI olmak üzere 4 adet kolon bulunmakta, bu kolonlara göre devamsızlık verileri tutulmaktadır. Bu veriler sayesinde hangi öğrencinin hangi derste yoklama bilgisi hakkında bilgilere ulaşabilirsiniz. Bir öğrencinin birden fazla derste devamsızlık durumu olduğu için bir e çok ilişkileri vardır. Öğrenci ve DERS tablolarından faydalanılarak OGRENCI_ID, DERS_ID kolonları ile eşleyip öğrencilerin ve derslerin diğer bilgilerine de ulaşabileceksiniz.

19)Bölümler: Tablomda ID, BÖLÜM_ADI olmak üzere 2 adet kolon bulunmakta, bu kolonlara göre bölümlerin verileri tutulmaktadır. Bu veriler sayesinde okulda hangi bölümler olduğu ve sayılarının ne kadar olduğu hakkında bilgilere ulaşabilirsiniz.

20)Faturalar: Tablomda BELGE_NO, OGRENCI_ID, BELGE_TARIHI, TUTAR olmak üzere 4 adet kolon bulunmakta, bu kolonlara göre faturaların verileri tutulmaktadır. Bu veriler sayesinde hangi öğrencinin hangi tarihte okula ne kadar ödediği hakkında bilgilere ulaşabilirsiniz. Bir e çok ilişki vardır. Öğrenci tablosundan faydalanılarak OGRENCI_ID kolonu ile eşleyip öğrencilerin diğer bilgilerine de ulaşabileceksiniz.

TABLO:



SQL ve İLİŞKİSEL CEBİR

1. Hes koduna göre riskli gözükten öğrencilerin adı soyadı ile birlikte bulunuz.

-SQL

```
SELECT OGRENCI_NUMARASI, AD, SOYAD, DURUM
FROM
A_OGRENCI O
JOIN
A_HESKODU H ON O.OGRENCI_NUMARASI = H.OGRENCI_ID
WHERE
DURUM = 'RİSKLİ'
```

-İLİŞKİSEL CEBİR

$\Pi_{AD,SOYAD,OGRENCI_NUMARASI,DURUM}(\sigma_{DURUM='Riskli'}(\text{Öğrenci} \bowtie \text{HesKodu.OGRENCI_ID}=\text{Öğrenci.OGRENCI_NUMARASI} \text{ HesKodu}))$

2. Ders programına göre öğrencilerin haftanın her gününde kaçar saat derse girdiklerini hesaplayınız.

-SQL

```
SELECT OGRENCI_NUMARASI, DERS_GUNU, COUNT(*)TOPLAM_DERS_SAYISI
FROM
A_DERSPROGRAMLARI DP
JOIN
A_OGRENCI O ON DP.OGRENCI_ID = O.OGRENCI_NUMARASI
JOIN
A_DERS D ON DP.DERS_ID = D.DERSKODU
GROUP BY
OGRENCI_NUMARASI,DERS_GUNU
ORDER BY
OGRENCI_NUMARASI DESC
```

-İLiSKİSEL CEBİR

Π OGRENCI_NUMARASI,DERS_GUNU,COUNT(*)((Öğrenci \bowtie DersProgramı.OGRENCI_ID=Öğrenci .OGRENCI_NUMARASI DersProgramı)U(DersProgramı \bowtie Ders.DERS_KODU= DersProgramı.DERS_ID Ders))

3. Devamsızlık bilgilerine göre öğrencilerin derslerdeki devam oranı %50'nin altına düştüğünde bir alarm döndürüp o öğrencinin o dersten devamlılık olarak kalabileceğini ifade ediniz.

-SQL

```
SELECT OGRENCI_NUMARASI,DERS_ADİ,DEVAM_ORANI,
CASE
WHEN
DEVAM_ORANI <= 50 THEN 'Riskli'
ELSE
'Risksiz'
END DURUM
FROM
A_DEVAMSIZLIK DV
JOIN
A_OGRENCI O ON DV.OGRENCI_ID = O.OGRENCI_NUMARASI
JOIN
A_DERS D ON DV.DERS_ID = D.DERSKODU
```

-İLiSKİSEL CEBİR

Π OGRENCI_NUMARASI,DERS_ADİ,DEVAM_ORANI,DURUM((Öğrenci \bowtie Devamsızlık.OGRENCI_ID=Öğrenci .OGRENCI_NUMARASI Devamsızlık)U(Devamsızlık \bowtie Ders.DERS_KODU= Devamsızlık.DERS_ID Ders))

PL-SQL PAKET VE FONKSİYONLAR

Paket oluşturdum ve içinde öğrencilerin staj türleri, devamlılık bilgileri ve mailleri hakkında fonksiyonlar tanımladım. Örnek olarak devamlılık_oranı fonksiyonunda öğrencinin aldığı tüm derslere bakılarak devam oranı %50'nin altında ise o öğrencinin o dersin yoklamasından kalabileceğini gösterecek.

```
CREATE OR REPLACE PACKAGE KISISEL_NOT_TAKIBI AS
    FUNCTION GONULLU_STAJ_TURU_SAYISI(OGR_ID PLS_INTEGER)
    RETURN PLS_INTEGER;
    FUNCTION ZORUNLU_STAJ_TURU_SAYISI(OGR_ID PLS_INTEGER)
    RETURN PLS_INTEGER;
    FUNCTION DEVAMLILIK_ORANI(OGR_ID PLS_INTEGER)
    RETURN VARCHAR2;
    FUNCTION MAILLER(OGR_ID PLS_INTEGER)
    RETURN VARCHAR2;
END KISISEL_NOT_TAKIBI;
/
```

```
CREATE OR REPLACE PACKAGE BODY KISISEL_NOT_TAKIBI IS
    FUNCTION GONULLU_STAJ_TURU_SAYISI(OGR_ID PLS_INTEGER) RETURN PLS_INTEGER IS
        GONULLU_STAJ PLS_INTEGER;
    BEGIN
        SELECT COUNT(*) INTO GONULLU_STAJ FROM A_STAJLAR S
        JOIN
            A_OGRENCI O ON S.OGRENCI_ID = O.OGRENCI_NUMARASI
            WHERE OGRENCI_ID = OGR_ID
            AND STAJ_TURU = 'Gönüllü'
        GROUP BY STAJ_TURU;
        RETURN GONULLU_STAJ;
    END;
```

```
    FUNCTION ZORUNLU_STAJ_TURU_SAYISI(OGR_ID PLS_INTEGER) RETURN PLS_INTEGER IS
        ZORUNLU_STAJ PLS_INTEGER;
    BEGIN
        SELECT COUNT(*) INTO ZORUNLU_STAJ FROM A_STAJLAR S
        JOIN
            A_OGRENCI O ON S.OGRENCI_ID = O.OGRENCI_NUMARASI
            WHERE OGRENCI_ID = OGR_ID
            AND STAJ_TURU = 'Zorunlu'
        GROUP BY STAJ_TURU;
        RETURN ZORUNLU_STAJ;
    END;
```

```

FUNCTION DEVAMLILIK_ORANI(OGR_ID PLS_INTEGER) RETURN VARCHAR2 IS
DEVAM_ORANI VARCHAR2(100);

BEGIN
--RETURN(KATILIM_SAGLANAN_DERS_SAYISI*TOPLAM_DERS_SAYISI)/100;
SELECT
CASE
WHEN MIN(DEVAM_ORANI)<50 THEN CONCAT(A_DERS.DERS_ADI, ' DERSİNDEN DEVAMSIZLIKTAN KALDI')
ELSE CONCAT(A_DERS.DERS_ADI, ' DERSİNDEN DEVAMSIZLIKTAN KALMADI')
END DURUM INTO DEVAM_ORANI
FROM A_DEVAMSIZLIK D
JOIN
A_OGRENCI O ON D.OGRENCI_ID = O.OGRENCI_NUMARASI
JOIN
A_DERS ON D.DERS_ID = A_DERS.DERS_KODU
WHERE OGRENCI_ID = OGR_ID
GROUP BY DERS_ADI;
RETURN DEVAM_ORANI;
END;
```

```

FUNCTION MAILLER(OGR_ID PLS_INTEGER) RETURN VARCHAR2 IS
v_mail VARCHAR2(100);
BEGIN
SELECT
MAIL_ADRESI INTO v_mail
FROM A_MAILLER M
JOIN
A_OGRENCI O ON M.OGRENCI_ID = O.OGRENCI_NUMARASI
WHERE OGRENCI_NUMARASI = OGR_ID;
RETURN v_mail;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RETURN NULL;
WHEN TOO_MANY_ROWS THEN
RETURN NULL;
END;
END KISISEL_NOT_TAKIBI;
/
```



```
declare
    ZORUNLU_STAJ_SAYISI PLS_INTEGER;
    MAIL VARCHAR2(100);
    GONULLU_STAJ_SAYISI PLS_INTEGER;
    ORAN VARCHAR2(100);
begin
    GONULLU_STAJ_SAYISI:=KISISEL_NOT_TAKIBI.GONULLU_STAJ_TURU_SAYISI(30119);
    MAIL:=KISISEL_NOT_TAKIBI.MAILLER(30119);
    ORAN:=KISISEL_NOT_TAKIBI.DEVAMLILIK_ORANI(30120);
    ZORUNLU_STAJ_SAYISI:=KISISEL_NOT_TAKIBI.ZORUNLU_STAJ_TURU_SAYISI(30119);

    dbms_output.put_line('Gönüllü Staj Sasyısı: '||GONULLU_STAJ_SAYISI);
    dbms_output.put_line('MAIL: '||MAIL);
    dbms_output.put_line('Devamsızlık oranı: '||ORAN);
    dbms_output.put_line('Zorunlu Staj Sasyısı: '||ZORUNLU_STAJ_SAYISI);
end;
```

Script Output x Query Result x

Task completed in 0,016 seconds

Dbms Output x

Buffer Size: 20000

HR_XE x

Gönüllü Staj Sasyısı: 1
MAIL: halisfurkanakgul@gmail.com
Devamsızlık oranı: BP 1 DERSİNDEN DEVAMSIZLIKTAN KALDI
Zorunlu Staj Sasyısı: 1

```
DECLARE
CURSOR c_satir
IS
SELECT
    OGRENCI_NUMARASI,DERS_ADI,DERS_GUNU
FROM A_DERS D
JOIN
    A_DERSPROGRAMI DP ON D.DERS_KODU = DP.DERS_ID
JOIN
    A_OGRENCI O ON DP.OGRENCI_ID = O.OGRENCI_NUMARASI
WHERE OGRENCI_NUMARASI = 30118
group by OGRENCI_NUMARASI,DERS_ADI,DERS_GUNU
order by ders_gunu;
begin
for x in c_satir
loop
    dbms_output.put_line( x.OGRENCI_NUMARASI || ' Numaralı Öğrenci ' || x.DERS_GUNU || ' gününde ' || x.DERS_ADI || ' adlı dersi vardır. ');
end loop;
end;
```

Script Output x Query Result x

Task completed in 0,015 seconds

Dbms Output x

Buffer Size: 20000

HR_XE x

30118 Numaralı Öğrenci Çarşamba gününde Calculus 1 adlı dersi vardır.
30118 Numaralı Öğrenci Salı gününde BP 2 adlı dersi vardır.
30118 Numaralı Öğrenci Salı gününde BP 1 adlı dersi vardır.

PL-SQL TRIGGER:

Yazmış olduğum trigger'ın amacı eğer sınav takvimi tablosunda şimdiki tarihin öncesinde bir sınavın tarihi varsa bunu bizlere getirecek.

```
CREATE OR REPLACE TRIGGER SilinenSınavTarihleri
BEFORE DELETE OR INSERT OR UPDATE
ON SınavTakvimi
FOR EACH ROW
BEGIN
    INSERT INTO SILINENSINAVTARİHLERİ
    (ID,
    OGRENCI_ID,
    DERS_ID,
    SINAV_TARİHİ,
    SINAV_YERİ)
VALUES
( :old.ID,
:old.OGRENCI_ID,
:old.DERS_ID,
:old.SINAV_TARİHİ,
:old.SINAV_YERİ);
END;
```

NORMALİZASYON:

1NF(1.NORMAL FORM):

Aşağıdaki tablo 1NF kurallarına uygun değildir. Çünkü aynı tablo içerisinde aynı isimde birden fazla kolon bulunamaz. (Aşağıdaki SINAV_YERİ ve SINAV_YERİ2 kolonları buna örnektir.) Her kolonda sadece tek bir değer bulunabilir.

ID	OGRENCI_ID	DERS_ID	SINAV_TARİHİ	SINAV_YERİ	SINAV_YERİ2	AD	SOYAD	BOLUM	DERS_ADI
1	30118	1	20.02.2022	A1	A2	TEST	DENEME	PC MÜH.	VERİ TABANI
2	30119	2	03.03.2022	C2	C3	TEST2	DENEME2	ELEK. MÜH.	FİZİK-2
3	30120	1	28.12.2021	B3	B4	TEST3	DENEME3	HUKUK K	HUKUK GİRİŞ

2NF(2.NORMAL FORM):

Aşağıdaki tablo 1NF kurallarına uygundur. Fakat 2NF kurallarına uygun değildir. Çünkü 2NF sadece composite primary key olan tablolara uygulanır. Herhangi bir veri alt kümesi birden fazla çok satırda tekrarlanmamalıdır. Ana tablo ile yan tablo arasında primary key-foreign key ilişkisi kurulmalıdır.

ID	OGRENCI_ID	DERS_ID	SINAV_TARIHI	SINAV_YERI	AD	SOYAD	BOLUM	DERS_ADI	AKTS
1	30118	1	20.02.2022	A1	TEST	DENEME	PC MÜH.	VERİ TABANI	4.0
2	30119	3	03.03.2022	C2	TEST2	DENEME2	ELEK. MÜH.	FİZİK-2	5.0
3	30120	6	28.12.2021	B3	TEST3	DENEME3	HUKUK	HUKUK GİRİŞ	4.0

R1={

ID, OGRENCI_ID, DERS_ID, SINAV_TARIHI, SINAV_YERI, AD, SOYAD, BOLUM, DERS_ADI, AKTS }

R2={

OGRENCI_ID -> AD, SOYAD, BOLUM, }

R3={

DERS_ID -> DERS_ADI,AKTS }

3NF(3.NORMAL FORM):

Aşağıdaki tablo 2NF kurallarına uygundur. Fakat 3NF kurallarına uygun değildir. Çünkü anahtar olmayan hiçbir kolon anahtar olmayan başka bir kolona bağlanmamalıdır.

ID	OGRENCI_ID	DERS_ID	SINAV_TARIHI	SINAV_YERI	AD	SOYAD	BOLUM	DERS_ADI	AKTS
1	30118	1	20.02.2022	A1	TEST	DENEME	PC MÜH.	VERİ TABANI	4.0
2	30119	3	03.03.2022	C2	TEST2	DENEME2	ELEK. MÜH.	FİZİK-2	5.0
3	30120	6	28.12.2021	B3	TEST3	DENEME3	HUKUK	HUKUK GİRİŞ	4.0
4	30120	3	03.03.2022	C2	TEST3	DENEME3	ELEK. MÜH.	FİZİK-2	5.0
5	30120	3	20.02.2022	A1	TEST3	DENEME3	PC MÜH.	VERİ TABANI	4.0

DERS_ID	DERS_ADI
1	VERİ TABANI
3	FİZİK-2
6	HUKUK GİRİŞ

OGRENCI_ID	AD	SOYAD
1821221018	TEST	DENEME
1821221019	TEST2	DENEME2
1821221020	TEST3	DENEME3

BCNF:

Aşağıdaki tablo 3NF kurallarına uygundur. Fakat BCNF kurallarına uygun değildir. Çünkü her belirleyici kolon (determinant), aynı zamanda bir aday anahtar olmalıdır.

OGRENCI_ID	DERS_ID	SINAV_TARIHI	SINAV_YERI	AD	SOYAD	BOLUM	DERS_ADI	AKTS
30118	1	20.02.2022	A1	TEST	DENEME	PC MÜH.	VERİ TABANI	4.0
30119	3	03.03.2022	C2	TEST2	DENEME2	ELEK. MÜH.	FİZİK-2	5.0
30120	6	28.12.2021	B3	TEST3	DENEME3	HUKUK	HUKUK GİRİŞ	4.0

OGRENCI_ID	DERS_ID
30118	1
30118	3
30118	6
30119	4
30119	6
30120	2

DERS_ID	DERS_ADI
1	VERİ TABANI
2	FİZİK-1
3	FİZİK-2
4	MİKROİŞLEMCİLER
6	HUKUK GİRİŞ

DENORMALİZASYON:

Veri tabanının performans artışı ve bazı işlemleri basitleştirmek için tekrar eden verinin tekrar sisteme eklenmesidir. Örnek olarak bazı sonuçlara ulaşmak için birden fazla tabloyu birleştirip o sonuca varmamız gerekir. Bunun sonucunda ise tabloları joinlediğimizden dolayı sorgumuz istediğimiz performansta çalışmayabilir. Bu sebepten dolayı denormalizasyon yapılmalıdır. Bu karmaşıklığı en aza indirgeyecektir. Veri tabanındaki fazlalıklar böylece en aza indirgenecek ve sorgumuz performanslı şekilde devam edecektir. Birden fazla denormalizasyon çeşidi vardır:

- Sorgu sonuçlarının saklanması
- Satır düzeyinde denormalizasyon
- Bileşik sorguları basitleştirmek
- Küme işlemlerinin sonuçlarının saklanması

Örnek olarak bir öğrencinin derslerdeki devamsızlık bilgilerini görmek için 3 tabloyu birbirleriyle join yaptığımız bir yapı olsun. Bu yapıyı başka projelerde, uygulamalarda vs. kullanacaksak bir tablo altında direkt olarak ulaşabilmemiz için create etmeliyiz. Yapılan bu işlem sonucunda tabloların her seferinde join olmasını ve sorgunun istediğimiz sonucu döndürmesini daha performanslı hale getirmiş olacağız.

Örnek ;

```
SELECT OGRENCI_NUMARASI,DERS_ADİ,DEVAM_ORANI,  
CASE  
WHEN  
DEVAM_ORANI <= 50 THEN 'Riskli'  
ELSE  
'Risksiz'  
END DURUM  
FROM  
A_DEVAMSIZLIK DV  
JOIN  
A_OGRENCI O ON DV.OGRENCI_ID = O.OGRENCI_NUMARASI  
JOIN  
A_DERS D ON DV.DERS_ID = D.DERSKODU
```

Yukarıdaki örnekte Devamsızlık, Öğrenci ve Ders tablolarını joinlememiz sayesinde sonuca ulaştığımızdır. Bunu her adımda yapmaktansa tabloya kaydetmek bize performans anlamında kazanç sağlayacaktır.