

```

CREATE TABLE calisanlar
(
id varchar(15) PRIMARY KEY,
isim varchar(30) UNIQUE,
maas int NOT NULL,
ise_baslama date
);
CREATE TABLE adresler
(
adres_id varchar(30),
sokak varchar(30),
cadde varchar(30),
sehir varchar(30),
FOREIGN KEY (adres_id) REFERENCES calisanlar(id)
);
INSERT INTO calisanlar VALUES('10002', 'Mehmet Yılmaz', 12000, '2018-04-14');
INSERT INTO calisanlar VALUES('10008', null, 5000, '2018-04-14');
INSERT INTO calisanlar VALUES('10010', 'Mehmet Yılmaz', 5000, '2018-04-14'); -- UNIQUE CONS. Kabul etmez
INSERT INTO calisanlar VALUES('10004', 'Veli Han', 5000, '2018-04-14');
INSERT INTO calisanlar VALUES('10005', 'Mustafa Ali', 5000, '2018-04-14');
INSERT INTO calisanlar VALUES('10006', 'Canan Yaş', NULL, '2019-04-12'); -- NOT NULL CONS. Kabul etmez
INSERT INTO calisanlar VALUES('10003', 'CAN', 5000, '2018-04-14');
INSERT INTO calisanlar VALUES('10007', 'CAN', 5000, '2018-04-14'); -- UNIQUE CONS. Kabul etmez
INSERT INTO calisanlar VALUES('10009', 'cem', '', '2018-04-14');
INSERT INTO calisanlar VALUES('', 'osman', 2000, '2018-04-14');
INSERT INTO calisanlar VALUES('', 'osman can', 2000, '2018-04-14'); -- PRIMARY KEY
INSERT INTO calisanlar VALUES('10002', 'ayse Yılmaz', 12000, '2018-04-14'); -- PRIMARY KEY
INSERT INTO calisanlar VALUES(null, 'filiz ', 12000, '2018-04-14'); -- PRIMARY KEY
INSERT INTO adresler VALUES('10003', 'Mutlu Sok', '40.Cad.', 'IST');
INSERT INTO adresler VALUES('10003', 'Can Sok', '50.Cad.', 'Ankara');
INSERT INTO adresler VALUES('10002', 'Ağa Sok', '30.Cad.', 'Antep');
-- Parent tabloda olmayan id ile child a ekleme yapamayiz
INSERT INTO adresler VALUES('10012', 'Ağa Sok', '30.Cad.', 'Antep'); -- id '10012' parent tabloda yok bu yüzden eklenemez hata verir
-- FK'ye null değeri atanabilir.
INSERT INTO adresler VALUES(NULL, 'Ağa Sok', '30.Cad.', 'Antep');
INSERT INTO adresler VALUES(NULL, 'Ağa Sok', '30.Cad.', 'Maraş');
SELECT * FROM calisanlar;
SELECT * FROM adresler;

```

Tabloya “CHECK Constraint” Nasıl Eklenir ?

CHECK ile bir alana girilebilecek değerleri sınırlandırabiliriz.

Mesela tablomuzda YAŞ bir alanı **number** data tipinde yani sayısal alan olarak belirlemiş olabiliriz. Ancak bu alan negatif sayı girilmesi anlamsız olacağı için CHECK yapısını kullanarak negatif giriş yapılmasını engelleyebiliriz.

Ornek : şehirler2 tablosu oluşturalım, nüfusun negatif değer girilmemesi için sınırlandırma (Constraint) koyalım

```
CREATE TABLE şehirler2
(
  alan_kodu int PRIMARY KEY,
  isim varchar(20) NOT NULL,
  nüfus int CHECK (nüfus>0)
);
```

Eğer (-) değer girersek



```
-- CHECK CONSTRAINT
```

```
CREATE TABLE calisanlar1
```

```
(
```

```
id varchar(15) PRIMARY KEY,
```

```
isim varchar(30) UNIQUE,
```

```
maas int CHECK (maas>10000),
```

```
ise_baslama date
```

```
);
```

```
INSERT INTO calisanlar1 VALUES('10052', 'Mehmet Yılmaz' ,9000, '2018-04-14');
```

CONSTRAINT

UNIQUE - Bir sütundaki tüm değerlerin BENZERSİZ/TEKRARSIZ yani tek olmasını sağlar.

NOT NULL - Bir Sütunun NULL içermemesini yani boş olmamasını sağlar.

NOT NULL kısıtlaması için constraint ismi tanımlanmaz. Bu kısıtlama veri türünden hemen sonra yerleştirilir

PRIMARY KEY - Bir sütünün NULL içermemesini ve sütundaki verilerin BENZERSİZ olmasını sağlar. (NOT NULL ve UNIQUE)

FOREIGN KEY - Başka bir tablodaki Primary Key'i referans göstermek için kullanılır. Böylelikle, tablolar arasında ilişki kurulmuş olur.

Check - Bir sütuna yerleştirilebilecek değer aralığını sınırlamak için kullanılır .

DQL - SELECT KOMUTU

Tablodaki Belli Bir Field'i Çağırma

SELECT isim

FROM calisanlar; Tablodan sadece isim field'indaki tüm dataları getirir

SELECT isim

FROM calisanlar

WHERE maas>5000; Tablodan sadece maas'ı 5000 den büyük olanların isim field'indaki dataları getirir

---DQL--WHERE KULLANIMI

SELECT * FROM calisanlar;

SELECT isim FROM calisanlar;

--calisanlar tablosundan maası 5000 den büyük olan isimleri listeleyiniz.

SELECT isim,maas FROM calisanlar WHERE maas>5000;

--Calisanlar Tablosundan İsmi Veli han olan veriyi getiriniz

SELECT*FROM calisanlar WHERE isim='Veli Han';

--calisanlar tablosundan maası 5000 olan tüm verileri listeleyiniz.

SELECT*FROM calisanlar WHERE maas=5000;

--DML --DELETE KOMUTU

DELETE FROM calisanlar;

DELETE FROM adresler WHERE sehir='Antep';

SELECT *FROM adresler;

```
--PRIMARY KEY OLUŞTURMA
```

```
CREATE TABLE ogrenciler8  
(  
ogrenci_no char(7) primary key,  
isim varchar(20),  
soyisim varchar(25),  
not_ort real,  
kayit_tarih date  
);
```

```
--PRIMARY KEY ATAMASI İKİNCİ YOL
```

```
CREATE TABLE ogrenciler9  
(  
ogrenci_no char(7),  
isim varchar(20),  
soyisim varchar(25),  
not_ort real,  
kayit_tarih date,  
CONSTRAINT ogr PRIMARY KEY (ogrenci_no)  
);
```