

BSM211 Veritabanı Yönetim Sistemleri - Celal ÇEKEN, İsmail ÖZTEL, Veysel Harun ŞAHİN

# Temel SQL (SQL DDL Komutları; CREATE, ALTER, DROP, ADD, SET, CONSTRAINTS)

---

## Konular

- Nesne oluşturma, silme, düzenleme
- SQL Kısıtları (CONSTRAINTS)

## Nesne oluşturma, silme, düzenleme

### CREATE

- Nesne (veritabanı, şema, tablo, view, fonksiyon vb.) oluşturmak için kullanılır

### CREATE DATABASE

- Veritabanı oluşturmak için kullanılır.

```
CREATE DATABASE "AlisVerisUygulamasi"  
ENCODING='UTF-8'  
LC_COLLATE='tr_TR.UTF-8' -- Bu özellik sonradan değiştirilemez (sıralama  
(karşılaştırma) işlemleri için)  
LC_CTYPE='tr_TR.UTF-8' -- Bu özellik sonradan değiştirilemez (büyük küçük harf  
dönüşümü, bir veri harf mi? vb.)  
OWNER postgres  
TEMPLATE=template0;
```

- Windows işletim sistemi için

```
CREATE DATABASE "AlisVerisUygulamasi"  
ENCODING='UTF-8'  
LC_COLLATE='Turkish_Turkey.1254'  
LC_CTYPE='Turkish_Turkey.1254'  
OWNER postgres  
TEMPLATE=template0;
```

### CREATE SCHEMA

- Veritabanını mantıksal olarak bölümlere ayırmak için kullanılır.
- Sabit disklerdeki klasör yapısına benzetilebilir. Bu sayede veritabanı daha kolay yönetilir. Çok sayıda kişinin aynı projede çalışabilmesi (isim uzayı) ve güvenlik kolaylaşır.

```
CREATE SCHEMA "sema1";
```

## CREATE TABLE

- Tablo oluşturmak için kullanılır.
- Tablo oluşturulurken sütunlarının veri tipi bildirilir.
- PostgreSQL'deki veri tiplerine aşağıdaki sayfadan erişilebilir.
  - <https://www.postgresql.org/docs/10/static/datatype.html>
- Sütunlar için uygun veri tipinin seçilmesi önemlidir. Bu yapıldığı takdirde;
  - Hız artar, kaynaklar etkin kullanılır.
  - Veriler tutarlı olarak saklanır (doğrulama).
  - Bazı saldırılara karşı önlem alınmış olur.

```
CREATE TABLE "sema1"."Urunler" (  
    "urunNo" SERIAL,  
    "kod" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktar" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kod"),  
    CONSTRAINT "urunlerCheck" CHECK("miktar" >= 0)  
);
```

## DROP TABLE

```
DROP TABLE "sema1"."Urunler";
```

## DROP SCHEMA

```
DROP SCHEMA "sema1";
```

## DROP DATABASE

```
DROP DATABASE "AlisVerisUygulamasi";
```

## TRUNCATE TABLE

- Bir tablonun içindeki tüm verileri silmek için kullanılır. Yapısal bir komuttur. Veriler gerçek anlamda silinir ve böylece işgal edilen yer sistem tarafından kullanılabilir hale gelir (Vacuum).

```
CREATE TABLE "Urunler" (  
    "urunNo" SERIAL,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
    CONSTRAINT "urunlerCheck" CHECK("miktarı" >= 0)  
);
```

```
INSERT INTO "Urunler"  
("kodu", "adi", "birimFiyati", "uretimTarihi", "miktarı")  
VALUES  
('ELO001', 'TV', 1300, '2019-10-30', 5);  
  
INSERT INTO "Urunler"  
("kodu", "adi", "birimFiyati", "uretimTarihi", "miktarı")  
VALUES  
('ELO002', 'TV', 1300, '2019-10-30', 5);
```

```
TRUNCATE TABLE "Urunler";
```

## ALTER TABLE

- Tablonun yapısını düzenlemek için kullanılır.

```
CREATE TABLE "Urunler" (  
    "urunNo" SERIAL,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
    CONSTRAINT "urunlerCheck" CHECK("miktarı" >= 0)  
);
```

```
ALTER TABLE "Urunler" ADD COLUMN "uretimYeri" VARCHAR(30);
```

```
ALTER TABLE "Urunler" DROP COLUMN "uretimYeri";
```

```
ALTER TABLE "Urunler" ADD "uretimYeri" VARCHAR(30);
```

```
ALTER TABLE "Urunler" ALTER COLUMN "uretimYeri" TYPE CHAR(20);
```

## SEQUENCE Nesnesi ve Değeri Otomatik Olarak Artan Alanlar

- Otomatik artım örneği - SERIAL kullanımı.

```
CREATE TABLE "Urunler" (  
    "urunNo" SERIAL,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
    CONSTRAINT "urunlerCheck" CHECK("miktarı" >= 0)  
);
```

```
INSERT INTO "Urunler"  
("kodu", "adi", "birimFiyati", "uretimTarihi", "miktarı")  
VALUES  
('EL0004', 'TV', 1300, '2019-10-30', 5);
```

- Otomatik artım örneği - SEQUENCE Kullanımı 1

```
CREATE TABLE "Urunler" (  
    "urunNo" INTEGER,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),
```

```
CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
CONSTRAINT "urunlerCheck" CHECK("miktar" >= 0)  
);
```

```
CREATE SEQUENCE "sayac";
```

```
ALTER SEQUENCE "sayac" OWNED BY "Urunler"."urunNo";
```

```
INSERT INTO "Urunler"  
("urunNo", "kodu", "adi", "birimFiyati", "uretimTarihi", "miktar")  
VALUES  
(NEXTVAL('"public"."sayac"'), 'EL0004', 'TV', 1300, '2019-10-30', 5);
```

- SEQUENCE nesnesinin bir sonraki değerini NEXTVAL kullanarak elde edebiliriz.
- SEQUENCE işleme fonksiyonlarını aşağıdaki bağlantıdan öğrenebiliriz.
- <https://www.postgresql.org/docs/10/static/functions-sequence.html>

```
SELECT NEXTVAL('sayac');
```

```
SELECT CURRVAL('sayac');
```

- Otomatik artım örneği - SEQUENCE Kullanımı 2
  - Tablo oluşturulurken de SEQUENCE kullanabiliriz.

```
CREATE SEQUENCE "sayac";
```

```
CREATE TABLE "Urunler" (  
  "urunNo" INTEGER DEFAULT NEXTVAL('sayac'),  
  "kodu" CHAR(6) NOT NULL,  
  "adi" VARCHAR(40) NOT NULL,  
  "uretimTarihi" DATE DEFAULT '2019-01-01',  
  "birimFiyati" MONEY,  
  "miktar" SMALLINT DEFAULT 0,  
  CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
  CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
  CONSTRAINT "urunlerCheck" CHECK("miktar" >= 0)  
);
```

```
INSERT INTO "Urunler"  
("urunNo", "kodu", "adi", "birimFiyati", "uretimTarihi", "miktar")  
VALUES  
(NEXTVAL('sayac'), 'EL0004', 'TV', 1300, '2019-10-30', 5);
```

```
INSERT INTO "Urunler"  
("kodu", "adi", "birimFiyati", "uretimTarihi", "miktar")  
VALUES  
('EL0004', 'TV', 1300, '2019-10-30', 5);
```

## SQL Kısıtları (CONSTRAINTS)

- Veri bütünlüğünün korunmasına yardımcı olur.

### NOT NULL

- Tanımlandığı alan boş olamaz. Veri girilmek zorundadır.

```
CREATE TABLE "Urunler" (  
    "urunNo" SERIAL,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktar" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
    CONSTRAINT "urunlerCheck" CHECK("miktar" >= 0)  
);
```

- "kodu" alanına veri girilmediği zaman hata alırsınız.

```
INSERT INTO "Urunler"  
("adi", "birimFiyati", "uretimTarihi", "miktar")  
VALUES  
('TV', 1300, '2019-10-30', 5);
```

```
ALTER TABLE "Urunler" ALTER COLUMN "kodu" DROP NOT NULL;
```

```
ALTER TABLE "Urunler" ALTER "kodu" SET NOT NULL;
```

## DEFAULT

- Tanımlandığı alana değer girilmemesi durumunda varsayılan bir değerin atanmasını sağlar.

```
CREATE TABLE "Urunler" (  
    "urunNo" SERIAL,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
    CONSTRAINT "urunlerCheck" CHECK("miktarı" >= 0)  
);
```

```
INSERT INTO "Urunler"  
("kodu", "adi", "birimFiyati", "miktarı")  
VALUES  
('ELO004', 'TV', 1300, 5);
```

```
ALTER TABLE "Urunler" ALTER "uretimTarihi" DROP DEFAULT;
```

```
ALTER TABLE "Urunler" ALTER COLUMN "uretimTarihi" SET DEFAULT '2019-01-01';
```

## UNIQUE

- Tanımlandığı alandaki verilerin eşsiz (tekil, benzersiz) olmasını sağlar.

```
CREATE TABLE "Urunler" (  
    "urunNo" SERIAL,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
    CONSTRAINT "urunlerCheck" CHECK("miktarı" >= 0)  
);
```

```
ALTER TABLE "Urunler" DROP CONSTRAINT "urunlerUnique";
```

```
ALTER TABLE "Urunler" ADD CONSTRAINT "urunlerUnique" UNIQUE ("kodu");
```

- İki alanlı UNIQUE örneği

```
CREATE TABLE "Urunler" (  
    "urunNo" SERIAL,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kodu", "adi"),  
    CONSTRAINT "urunlerCheck" CHECK("miktarı" >= 0)  
);
```

```
ALTER TABLE "Urunler" DROP CONSTRAINT "urunlerUnique";
```

```
ALTER TABLE "Urunler" ADD CONSTRAINT "urunlerUnique" UNIQUE ("kodu", "adi");
```

```
INSERT INTO "Urunler"  
("kodu", "adi", "birimFiyati", "uretimTarihi", "miktarı")  
VALUES  
('EL0004', 'TV', 1300, '2016-10-24', 5);
```

```
INSERT INTO "Urunler"  
("kodu", "adi", "birimFiyati", "uretimTarihi", "miktarı")  
VALUES  
('EL0005', 'Bilgisayar', 1500, '2019-10-20', 5);
```

## CHECK

- Tanımlandığı alandaki değer aralığını sınırlamada kullanılır.



```
CREATE TABLE "Urunler" (  
    "urunNo" SERIAL,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
    CONSTRAINT "urunlerCheck" CHECK("miktarı" >= 0)  
);
```

```
ALTER TABLE "Urunler" DROP CONSTRAINT "urunlerCheck";
```

```
ALTER TABLE "Urunler" ADD CONSTRAINT "urunlerCheck" CHECK ("miktarı" >= 0);
```

```
INSERT INTO "Urunler"  
("kodu", "adi", "birimFiyati", "uretimTarihi", "miktarı")  
VALUES  
('EL0004', 'Bilgisayar', 1300, '2016-04-05', -3);
```

### Birincil Anahtar (PRIMARY KEY)

```
CREATE TABLE "Urunler" (  
    "urunNo" SERIAL,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo")  
);
```

```
ALTER TABLE "Urunler" DROP CONSTRAINT "urunlerPK";
```

```
ALTER TABLE "Urunler" ADD CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo");
```

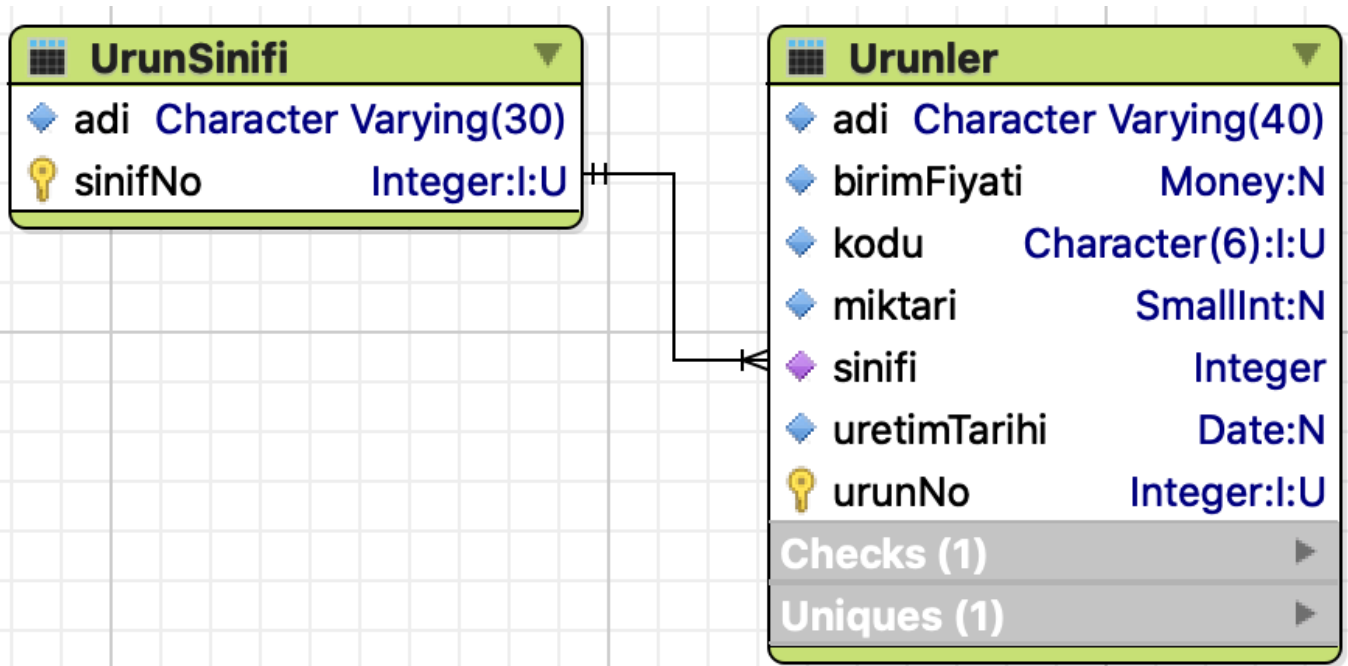
- İki alanlı birincil anahtar örneği.

```
CREATE TABLE "Urunler1" (
  "urunNo" SERIAL,
  "kodu" CHAR(6) NOT NULL,
  "adi" VARCHAR(40) NOT NULL,
  "uretimTarihi" DATE DEFAULT '2019-01-01',
  "birimFiyati" MONEY,
  "miktarı" SMALLINT DEFAULT 0,
  CONSTRAINT "urunlerPK1" PRIMARY KEY("urunNo", "kodu")
);
```

```
ALTER TABLE "Urunler1" DROP CONSTRAINT "urunlerPK1";
```

```
ALTER TABLE "Urunler1" ADD CONSTRAINT "urunlerPK1" PRIMARY KEY("urunNo", "kodu");
```

### Yabancı Anahtar (FOREIGN KEY)



```
CREATE TABLE "UrunSinifi" (
  "sinifNo" SERIAL,
  "adi" VARCHAR(30) NOT NULL,
  CONSTRAINT "urunSinifiPK" PRIMARY KEY("sinifNo")
);
```

```
CREATE SEQUENCE "sayac";
```

```
CREATE TABLE "Urunler" (  
    "urunNo" INTEGER DEFAULT NEXTVAL('sayac'),  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "sinifi" INTEGER NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
    CONSTRAINT "urunlerCheck" CHECK("miktarı" >= 0),  
    CONSTRAINT "urunSinifiFK1" FOREIGN KEY("sinifi") REFERENCES "UrunSinifi"  
    ("sinifNo")  
);
```

- Bu ifade yukarıdaki ile aynıdır. ON DELETE ve ON UPDATE durumunda ne yapılacağı belirtilmediğinde varsayılan olarak NO ACTION olur.

```
CREATE TABLE "Urunler" (  
    "urunNo" SERIAL,  
    "kodu" CHAR(6) NOT NULL,  
    "adi" VARCHAR(40) NOT NULL,  
    "urunTipi" INTEGER NOT NULL,  
    "uretimTarihi" DATE DEFAULT '2019-01-01',  
    "birimFiyati" MONEY,  
    "miktarı" SMALLINT DEFAULT 0,  
    CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),  
    CONSTRAINT "urunlerUnique" UNIQUE("kodu"),  
    CONSTRAINT "urunlerCheck" CHECK("miktarı" >= 0),  
    CONSTRAINT "urunlerFK" FOREIGN KEY("urunTipi") REFERENCES "UrunTipleri"  
    ("tipNo") ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

- Üç davranış şekli vardır: NO ACTION (varsayılan), RESTRICT, CASCADE

```
ALTER TABLE "Urunler" DROP CONSTRAINT "urunlerFK";
```

```
ALTER TABLE "Urunler"  
ADD CONSTRAINT "urunlerFK" FOREIGN KEY("urunTipi")  
REFERENCES "UrunTipleri"("tipNo")  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;
```

```
ALTER TABLE "Urunler"  
ADD CONSTRAINT "urunlerFK" FOREIGN KEY("urunTipi")  
REFERENCES "UrunTipleri"("tipNo")  
ON DELETE RESTRICT  
ON UPDATE RESTRICT;
```

```
ALTER TABLE "Urunler"  
ADD CONSTRAINT "urunlerFK" FOREIGN KEY("urunTipi")  
REFERENCES "UrunTipleri"("tipNo")  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```