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 (arama, sıralama 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,

"kodu" CHAR(6) NOT NULL,

"adi" VARCHAR(40) NOT NULL,

"uretimTarihi" DATE DEFAULT '2019-01-01',

"birimFiyati" MONEY,

"miktari" SMALLINT DEFAULT 0,

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 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.

CREATE TABLE "Urunler" (

"urunNo" SERIAL,

"kodu" CHAR(6) NOT NULL,

"adi" VARCHAR(40) NOT NULL,

"uretimTarihi" DATE DEFAULT '2019-01-01',

"birimFiyati" MONEY,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 0)

);

INSERT INTO "Urunler"

("kodu", "adi", "birimFiyati", "uretimTarihi", "miktari")

VALUES

('ELO001', 'TV', '13', '2019-10-30', 5);

INSERT INTO "Urunler"

("kodu", "adi", "birimFiyati", "uretimTarihi", "miktari")

VALUES

('ELO002', 'TV', '13', '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,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 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);

* 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,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 0)

);

INSERT INTO "Urunler"

("kodu", "adi", "birimFiyati", "uretimTarihi", "miktari")

VALUES

('ELO004', 'TV', '13', '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,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 0)

);

CREATE SEQUENCE "sayac";

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

INSERT INTO "Urunler"

("urunNo", "kodu", "adi", "birimFiyati", "uretimTarihi", "miktari")

VALUES

(NEXTVAL('"public"."sayac"'), 'ELO004', 'TV', '13', '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,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 0)

);

INSERT INTO "Urunler"

("urunNo", "kodu", "adi", "birimFiyati", "uretimTarihi", "miktari")

VALUES

(NEXTVAL('sayac'), 'ELO004', 'TV', '13', '2019-10-30', 5);

INSERT INTO "Urunler"

("kodu", "adi", "birimFiyati", "uretimTarihi", "miktari")

VALUES

('ELO004', 'TV', '13', '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,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 0)

);

* "kodu" alanına veri girilmediği zaman hata alırız.

INSERT INTO "Urunler"

("adi", "birimFiyati", "uretimTarihi", "miktari")

VALUES

('TV', '13', '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,

"miktari" SMALLINT DEFAULT 0,

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 0)

);

INSERT INTO "Urunler"

("kodu", "adi", "birimFiyati", "miktari")

VALUES

('ELO004', 'TV', '13', 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,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 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,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu", "adi"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 0)

);

ALTER TABLE "Urunler" DROP CONSTRAINT "urunlerUnique";

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

INSERT INTO "Urunler"

("kodu", "adi", "birimFiyati", "uretimTarihi", "miktari")

VALUES

('ELO004', 'TV', '13', '2016-10-24', 5);

INSERT INTO "Urunler"

("kodu", "adi", "birimFiyati", "uretimTarihi", "miktari")

VALUES

('ELO005', 'Bilgisayar', '13', '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,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 0)

);

ALTER TABLE "Urunler" DROP CONSTRAINT "urunlerCheck";

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

INSERT INTO "Urunler"

("kodu", "adi", "birimFiyati", "uretimTarihi", "miktari")

VALUES

('ELO004', 'Bilgisayar', '13', '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,

"miktari" 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,

"miktari" 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 "UrunTipleri" (

"tipNo" SERIAL,

"adi" VARCHAR(30) NOT NULL,

CONSTRAINT "urunTipiPK" PRIMARY KEY("tipNo")

);

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,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 0),

CONSTRAINT "urunlerFK" FOREIGN KEY("urunTipi") REFERENCES "UrunTipleri"("tipNo")

);

* 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,

"miktari" SMALLINT DEFAULT '0',

CONSTRAINT "urunlerPK" PRIMARY KEY("urunNo"),

CONSTRAINT "urunlerUnique" UNIQUE("kodu"),

CONSTRAINT "urunlerCheck" CHECK("miktari" >= 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;