VERİTABANI YÖNETIM SİSTEMLERİ

Crate, Insert, Alter

- ***MS SQL Server*** ile oluşturulan veritabanı dosyalarının uzantısı mdf ve buna bağlı log dosyasının uzantısı da ldf(lod data file)'dir. "mdf" uzantılı dosya verilerin fiziksel olarak tutulduğu veritabanı dosyasının kendisidir. "ldf" uzantılı dosya ise veritabanında meydana gelen değişiklikleri tutan işlem günlüğü dosyasını içerir. Transaction log (işlem günlüğü) dosyaları, SQL Server içinde, veritabanında meydana gelen bütün değişiklikleri saklar. Bütün düzeltme işlemleri, veritabanına yazılmadan önce log dosyalarına yazılır. Bu sayede, veritabanının içerdiği verilerin kaybolmasına neden olacak bir durum gerçekleştiğinde, bu dosyalar veritabanı kurtarma işlemlerinde kullanılabilir.

CREATE (YARAT) KOMUTU

- Database, tablo, index, view vb. veri tabanı objelerini yaratmada kullanılan komuttur. SQL komutları ile veri tabanında işlem yapılabilmesi için önce veri tabanı sonra da veri tabanında kullanılacak tablolar tanımlanmalıdır. SQL Server'da veritabanı oluşturmak için CREATE DATABASE deyimi kullanılır.

- Kullanılışı: CREATE DATABASE isim;
- Bu komut isim ile belirtilen isimde bir veri tabanı oluşturur. Bir veri tabanı içerisinde çok sayıda veri tabanı kütüğü ya da tablo bulunabilir. Her tabloda saklanan verilerle ilgili alanlar bulunmalıdır.

- **Örnek:** Deneme, okul, sirket adında veritabanı oluşturan komutları yazınız.
- Create Database deneme;
- Create Database okul;
- Create Database sirket;

VERİ TABANINI AKTİF YAPMA

- USE komutu kullanılır. Önceden yapılmış veri tabanını aktif hale getirir.

- **Örnek:** Deneme, okul ve sirket adında oluşturduğunuz veritabanını aktif yapan komutları yazınız.
- Use deneme;
- Use okul;
- Use sirket;

TABLO OLUŞTURMA

CREATE TABLE deyimi kullanılır. Kullanılacak tüm alanlar bu deyim içerisinde belirtilmelidir.

Örnek: Yukarı da bahsedilen örnek okul veritabanı projesinin veritabanını oluşturup, aktif yapıp, bahsedilen tabloları oluşturan komutları yazınız.(Tablo 5.1, 5.2, 5.3'den yararlanarak)

```
Create Database proje
Use proje
Create table bolum(bolkod smallint primary key , bol_adi char(15))
Create table unvan(un_kod tinyint primary key , ünvani char(11))
Create table hocalar(h_ID tinyint primary key , h_adi char(20) , h_soyadi char(20) , unvan tinyint foreign key references unvan(un_kod) , top_ders_saati tinyint)
Create table dersler(op_kod smallint primary key , ders_kod char(10) , ders_adi char(20) , dönem tinyint , teori tinyint , pratik tinyint , hocasi tinyint foreign key references hocalar(h_ID))
Create table notlar(no int , op_kod smallint foreign key references dersler(op_kod) , vize tinyint , check(vize between 0 and 100) , final tinyint , check(final between 0 and 100) , but tinyint , check(but between 0 and 100) )
Create table memleket(tr_kod tinyint primary key , m_adi char(15) , tel_kod smallint)
Create table ogrenci(no int primary key , adi char(20) , soyadi char(20) , bolkod smallint foreign key references bolum(bolkod) , sinif tinyint , check( sinif IN(1,2,3,4)) , h_ID tinyint foreign key references hocalar(h_ID) , d_tarihi datetime , memleket tinyint foreign key references memleket(tr_kod) , cinsiyet char(5) )
```

Benzersiz alan UNIQUE

- Veritabanı sistemlerinde veri bütünlüğünü sağlamak için **CONSTRAINT** olarak isimlendirilen bazı zorlayıcı ifadelerde kullanılabilir. Bunlar NULL/NOT NULL, PRIMARY KEY, FOREIGN KEY, CHECK, UNIQUE ve DEFAULT ifadeleridir.

```
CREATE TABLE personel(
personel_id int NOT NULL,
soyad varchar(255) NOT NULL,
 isim varchar(255),
 adres varchar(255),
 şehir varchar(255),
UNIQUE (personel_id))
```

otomatik artış **IDENTITY**

- ***MS SQL Server'da*** otomatik artış için **IDENTITY** (1,1) ifadesi kullanılır. Bu ifade artış birden başlasın ve birer birer artsın anlamına gelmektedir. Aşağıda kullanımıyla ilgli örnek verilmiştir.

```
CREATE TABLE öğrenciler(
Ogr_no int IDENTITY(1,1) NOT NULL,
ad nchar(10)
)
```

DEFAULT değer kullanımı

- tablo içerisinde veri girişi yapılmadan NULL olarak bırakılan sütunlara otomatik olarak veri girişi için kullanılır. Örneğin, Öğrenci bilgileri girilirken şehir bilgisi boş bırakıldığında bu alana otomatik olarak Tokat bilgisinin gelmesi sağlanabilir.
- CREATE TABLE öğrenciler(
- Ogr_no int NOT NULL, ad nchar(10),
- şehir nchar(50) DEFAULT 'TOKAT'
-)

PRIMARY KEY kullanımı

- Daha önceki uygulama içerisinde kullanılmıştı ama bazı durumlarda birincil anahtar olarak belirlenecek sütun iki sütununun birleşiminden oluşabilir. Örneğin, öğrencilerin gördüğü derslerin notlarının tutulduğu notlar tablosu için birincil anahtar ogr_no ve ders_kodu olarak belirlenebilir.

```
CREATE TABLE notlar(
      ogr_no int NOT NULL,
      ders_kodu varchar(25) NOT NULL,
      vize int,
      final int,
      CONSTRAINT pkkey PRIMARY KEY(ogr_no,ders_kodu))
```

	Column Name	Data Type	Allow Nulls
🔑	ogr_no	int	☐
🔑	ders_kodu	varchar(25)	☐
	vize	int	☑
	final	int	☑

CHECK kullanımı

- Tablo oluşturulurken bir sütun içerisine girilebilecek değerler için bir kısıtlama getirmek için kullanılır. Örneğin, öğrencinin not bilgisinin 0'dan küçük değer girilmesinin engellenmesi. Sütun için belirlenecek kontrol ifadesi tek bir koşul içeriyorsa(ör, 0'dan büyük olması) CHECK ifadesi sütun isminden sonra yazılarak koşul ifadesi verilir.

```
CREATE TABLE notlar (
        ogr_no int NOT NULL,
        ders_kodu varchar(25) NOT NULL,
        vize int CHECK (vize>=0),
        final int CHECK (final>=0),
        CONSTRAINT pkkey PRIMARY KEY(ogr_no,ders_kodu))
```

CHECK kullanımı

- Koşul ifadesi birden fazla olduğu durumlarda CHECK ifadesi CONTSRAINT olarak kullanılır. Aşağıda kullanım şekli verilmiştir.

```
CREATE TABLE notlar(
ogr_no int NOT NULL,
ders_kodu varchar(25) NOT NULL,
vize int,
final int,
CONSTRAINT chkvize CHECK (vize>=0 AND vize<=100),
CONSTRAINT chkfinal CHECK (final>=0 AND final<=100))
```

CHECK kullanımı

```
CREATE TABLE urunler (
urun_kod varchar(4) NOT NULL,
 urun_ad varchar(25) NOT NULL,
 CONSTRAINT chkkod CHECK (
urun_kod IN ('A089','A090','A010','A100') OR
urun_kod LIKE 'A9[0-9][0-9]'))
```

Bu şekilde bir kullanım için girilecek ürün kodu A089, A090, A010, A100 değerlerinden biri veya A900 ile A999 arasında olmak zorundadır. Farklı bir değer girildiğinde kayıt işlemi gerçekleşmeyecektir.

FOREIGN **KEY**

- 2 türlü kullanımı vardır.

```
CREATE TABLE notlar(
ogr_no int FOREIGN KEY REFERENCES ogrenci(ogr_no),
ders_kodu varchar(25) FOREIGN KEY REFERENCES
ders(ders_kodu),
 vize int,
final int)
CREATE TABLE notlar(
ogr_no int,
ders_kodu varchar(25),
vize int,
final int
CONSTRAINT fk_ogrno FOREIGN KEY(ogr_no) REFERENCES
span>renci(ogr_no),
CONSTRAINT fk_dkod FOREIGN KEY(derskodu) REFERENCES
ders(ders_kodu))
```

Uygulama

- Bir araç kiralama veritabanı için aşağıdaki tablolar verilmiştir. Bu tabloları oluşturmak için gerekli SQL ifadelerini yazalım.
- MUSTERI (m kod, mad, msoyad, madres, mtel)
- KIRALAMA (mkod. aracno. tarih, saat, tes_tarihi, tes_saati)
- ARAC (arac no. model, marka, plaka, fiyat)
- Verilen tablolar incelendiğinde altı çizgili olarak verilen sütunlar birincil anahtar(PRIMARY KEY) olarak belirlenecektir.
- Aynı zamanda KİRALAMA tablosunun mkod ve aracno sütunları MÜŞTERİ ve ARAÇ tablosunun birincil anahtar sütunları ve kendisinin FOREIGN KEY sütunudur.
- KİRALAMA tablosunun mkod ve aracno sütununun birleşimi de KİRALAMA tablosunun birincil anahtarını oluşturmaktadır.
- ARAÇ tablosunda bulunan model sütunu için girilecek bilgi 1900-2015 arasında olacak şekilde sınırlanmalıdır.

Uygulama

```
CREATE DATABASE uygulama2;
use uygulama2;

CREATE TABLE müşteri(
mkod int NOT NULL PRIMARY KEY,
 mad varchar(50) NOT NULL,
 msoyad varchar(50) NOT NULL,
 madres varchar(255),
 mtel varchar(15))
CREATE TABLE araç(
aracno int NOT NULL PRIMARY KEY,
 model int NOT NULL,
 marka varchar(50),
 plaka varchar(25),
 fiyat varchar(15),
CONSTRAINT chkmodel CHECK (model LIKE '[1-2][0-9][0-9][0-
9]'))
```

Uygulama

```
CREATE TABLE kiralama(
mkod int NOT NULL,
aracno int NOT NULL,
tarih varchar(10),
 saat varchar(8),
 tes_tarih varchar(10),
 tes_saat varchar(8),
CONSTRAINT fk_mkod FOREIGN KEY(mkod)
REFERENCES müşteri(mkod),
CONSTRAINT fk_aracno FOREIGN KEY(aracno)
REFERENCES araç(aracno),
CONSTRAINT pkkey PRIMARY KEY(mkod,aracno) )
```

FOREIGN KEY

- Tablolar arasında oluşturulan FOREIGN KEY ilişkiler referans olarak kullanılan tablolardan kayıt silme ve güncelleme işleminde ilişkilerden dolayı hata vermesine neden olacaktır. Örneğin, bir önceki uygulamada müşteri tablosuna 1 müşteri koduyla kaydedilen bir müşteriye araç kiralaması yapıldığını düşünelim. Daha sonra müşteri tablosundan 1 olan müşteri kodunu değiştirmek istediğimizde hatayla karşılaşacağız. Çünkü müşteri tablosunun mkod sütunu kiralama tablosunda FOREIGN KEY olarak kullanılmıştır. Bu tür sorunların engellenmesi için FOREIGN KEY ilişkiler oluşturulurken güncelleme veya silme işlemi için izin verilmelidir.

CASCADE

- FOREIGN KEY oluşturulurken **ON** DELETE CASCADE ifadesi kullanılırsa referans tablodan silinen satırın kullanıldığı diğer tablolardan da ilgili satırlar silinecektir. Aynı şeklide FOREIGN KEY oluşturulurken **ON UPDATE CASCADE** ifadesi kullanılırsa referans tabloda güncellenen satır kullanıldığı diğer tablolarda da otomatik olarak güncellenecektir.

CASCADE

- CONSTRAINT fk_mkod FOREIGN KEY(mkod) REFERENCES müşteri(mkod) ON UPDATE CASCADE ON DELETE CASCADE,
- CONSTRAINT fk_aracno FOREIGN KEY(aracno) REFERENCES araç(aracno) ON UPDATE CASCADE ON DELETE CASCADE,

- Örneğin, "müşteri" tablosunda bulunan 1 müşteri kodlu müşteriye araç kiralaması yapıldığını düşünelim. Daha sonra "müşteri" tablosundan 1 olan müşteri kodu 2 olarak değiştirilirse otomatik olarak "kiralama" tablosunda bulunan 1 müşteri kodları da 2 olacaktır. Aynı şekilde 1 kodlu müşteri "müşteri" tablosundan silinirse "kiralama" tablosundan da 1 nolu müşteriye yapılan kiralamalar silinecektir.

ALTER

- Daha önce oluşturulmuş veritabanı nesnesinin özelliğini değiştirmek için kullanılır. Yapılmak istenen değişiklik parametre olarak verilir.
- Aşağıda kullanım şekilleri verilmiştir.
- *ALTER TABLE* ***tablo*** *ADD* ***sütun_adı özellikler***
- Yukarıda verilen kullanım şekli belirtilen tabloya yeni bir sütun eklemek için kullanılır. Tablo oluştururken kullanılan NOT NULL ve UNIQUE gibi ifadeler aynı şekilde geçerlidir. Ayrıca ADD ifadesinden sonra, sütun adı ve özellikler yerine tablo oluşturmada anlatılan CONTRAINT ifadeler de kullanılabilir. Bu durumda belirtilen CONSTRAINT ifade tablodan silinecektir.
- **ALTER TABLE** *tablo* **DROP COLUMN** *sütun_adl*
- Yukarıda verilen kullanım şekli belirtilen tablodaki belirtilen sütunu silmek için kullanılır. Ayrıca DROP ifadesinden sonra, column, sütun adı ve özellikler yerine tablo oluşturmada anlatılan CONTRAINT ifadeler de kullanılabilir. Bu durumda belirtilen CONSTRAINT ifade tablodan silinecektir.
- **ALTER TABLE** *tablo* **ALTER COLUMN** *sütun_adı özellikler*
- Yukarıda verilen kulanım şekli belirtilen tablodaki belirtilen sütunun özelliklerini değiştirmek için kullanılır. Burada belirtilecek özellikler tablo oluşturmada kullanılanlarla aynıdır.

- **ALTER TABLE öğrenci ADD UNIQUE** (**ogr no**) şeklindeki kullanım var olan öğrenci tablosu için ogr_no sütununa UNIQUE özelliği eklemektedir.
- **ALTER TABLE öğrenci ADD CONSTRAINT UC_de**
- **UNIQUE** (**ogr_no, ad**) şeklindeki kullanım var olan öğrenci tablosu için ***ogr_no*** ve ad sütunlarına UNIQUE özelliği ekler.
- **ALTER TABLE öğrenci ADD PRIMARY KEY** (**ogr_no**) şeklindeki kullanım var olan öğrenci tablosu için ogr_no sütununa PRIMARY KEY özelliği ekler.
- ALTER **TABLE öğrenci DROP PRIMARY KEY** şeklindeki kullanım öğrenci tablosunun birincil anahtar özelliğini silecektir.

DROP İfadesi

- Veritabanı içerisinde var olan nesneleri veya veritabanını silmek için kullanılır.
- **DROP TABLE** *tablo_adl*
- Yukarıdaki kullanım tablo_adı bölümünde belirlenen tablonun silinmesi için kullanılır.
- **DROP DATABASE *veritabanı adı***

TRUNCATE

- Veri silmek için kullanılan bir diğer SQL ifadesi TRUNCATE TABLE'dır.
- Örnek: Aşağıdaki SQL ifadesi öğrenci tablosu içerisindeki tüm kayıtları silecektir. **TRUNCATE TABLE** öğrenci

TABLOLARA VERİ YÜKLENMESİ

Bir tabloya veri girişi (ya da veri yüklenmesi) işlemi için, SQL'de mevcut komut INSERT INTO / VALUES komutudur.

Örnek: Yukarıdaki örnekte oluşturduğunuz tablolara birer kayıt giriniz.

```
insert into bolum values(536,'Yapı-Resim');
```

Tablo 6.1: Bölüm Tablosu

	bolkod	bol_adi
	530	Elektronik
	531	Bilgisayar
	532	Metal
	533	Talaşlı
	534	Makine
	535	Yapı
	536	Yapı-Resim

```
insert into unvan values(1,'Prof.Dr.');
```

Tablo 6.2: Ünvan Tablosu

un_kod	ünvani
1	Prof.Dr.
2	Doç.Dr.
3	Yrd.Doç.Dr.
4	Öğr.GÖr.
5	Öğr.Gör.Dr.
6	Arş.Gör.

```
insert into dersler values(101,'tde 102','Türkdili',2,2,2,12)
```

Tablo 6.3: Dersler Tablosu

op_kod	ders_kod	ders_adi	dönem	teori	pratik	hocası
101	tde 102	Türkdili	2	2	2	12
104	tdı 132	İngilizce	2	2	0	16
114	bil 166	Nesne yön.prg.gr	2	2	2	14
115	bil 168	Bilgisayar donanımı	2	2	0	5
118	egt 162	Okul deneyimi	2	1	4	3
151	tde 101	Türk dili	1	2	0	12
155	bil 161	Temel bil.tek.kul.	1	2	2	11
164	mat 167	Matematik	1	3	0	12
167	egt 171	Öğrt.mesleğine giriş	1	3	0	3
218	elt 292	Elk.devreler	2	3	2	6
219	bil 282	Meslek mat.	2	3	0	1
222	bil 288	Bil.prg.2	2	2	2	13
270	bil 285	Mantık devreler	1	3	0	4
271	bil 287	Bil.prg.	1	3	2	15
274	elt 291	Devre analizi	1	3	2	7
317	egt 372	Sınıf yönetimi	2	2	2	8
324	bil 384	Çoklu ortam	2	2	2	11
326	bil 388	SQL	2	3	2	14
371	bil 383	Teknik iletişim	1	2	0	11
372	bil 385	Mikro. mimarisi	1	3	2	5
373	bil 387	Veri yapıları	1	3	0	13
374	elt 395	Ölçme	1	2	2	6
375	bil 391	İşletim sistemleri	1	3	2	10
421	egt 472	Rehberlik	2	3	0	9
433	egt 484	Bil.ağları	2	3	0	8
434	egt 486	Web Prog.	2	3	2	13
436	egt 490	Proje 2	2	0	2	8
467	egt 471	Okul deneyimi	1	1	4	9
481	egt 483	Bil.ağları2	1	3	2	1
482	egt 485	Web tasarımı	1	3	2	14
483	egt 487	Yapay zeka	1	3	0	13
484	egt 489	Proje	1	0	2	8

```
insert into hocalar values(1,'Asaf','Varol',1,20)
```

Tablo 6.4: Hocalar Tablosu

h_ID	h_adi	h_soyadi	unvan	top_ders_saati
1	Asaf	Varol	1	20
2	Muammer	Gökbulut	1	20
3	Hakan	Akpolat	1	20
4	Hanifi	Güldemir	1	20
5	İbrahim	Türkoğlu	2	18
6	Mehmet	Gedikpınar	3	17
7	Servet	Tuncer	3	17
8	Engin	Avcı	3	17
9	Davut	Hanbay	3	17
10	Abdülkadir	Şengür	3	17
11	Nurhayat	Varol	4	15
12	Cafer	Bal	5	16
13	Erkan	Tanyıldızı	5	16
14	Murat	Karabatak	4	15
15	Ferhat	Bahçacı	6	12
16	Korhan	Kayışlı	6	12

```
insert into memleket values(23,'Elazıg',424)
```

Tablo 6.5: Memleket Tablosu

tr_kod	m_adi	tel_kod
1	Adana	322
2	Adıyaman	416
3	Afyon	272
4	Ağrı	472
5	Amasya	382
6	Ankara	312
7	Antalya	242
8	Artvin	466
9	Aydın	256
10	Balıkesir	266
11	Bilecik	228
12	Bingöl	426
13	Bitlis	434
14	Bolu	374
15	Burdur	248
16	Bursa	224
17	Çanakkale	286
18	Çankırı	376
19	Çorum	364
20	Denizli	258
21	Diyarbakır	412
22	Edirne	284
23	Elazığ	424
24	Erzincan	446
25	Erzurum	442
26	Eskişehir	222
27	Gaziantep	342
28	Giresun	454
29	Gümüşh…	456
30	Hakkari	438
31	Hatay	326
32	Isparta	246
33	İçel	324
34	İstanbul	216
35	İzmir	232
36	Kars	474
37	Kastamo…	366
38	Kayseri	352
39	Kırklareli	288
40	Kırşehir	386
41	Koceli	262
42	Konya	332
43	Kütahya	247
44	Malatya	422
45	Manisa	236
46	K.Maraş	344
47	Mardin	482
48	Muğla	252
49	Muş	436
50	Nevşehir	384
51	Niğde	388
52	Ordu	452
53	Rize	464
54	Sakarya	264
55	Samsun	362
56	Siirt	484
57	Sinop	368
58	Sivas	346
59	Tekirdağ	282
60	Tokat	356
61	Trabzon	462
62	Tunceli	428
63	Şanlıurfa	414
64	Uşak	276
65	Van	432
66	Yozgat	354
67	Zonguldak	372
68	Aksaray	382
69	Bayburt	458
70	Karaman	338
71	Kırıkkale	318
72	Batman	488
73	Şırnak	486
74	Bartın	378
75	Ardahan	478
76	Iğdır	476
77	Yalova	226
78	Karabük	370
79	Kilis	348
80	Osmaniye	328
81	Düzce	111

```
insert notlar values(7536511,274,53,80,40)
```

Tablo 6.6: Notlar Tablosu

no	op_kod	vize	final	büt
7536511	274	53	80	40
7536511	218	98	76	54
7536521	274	21	78	63
7536521	222	57	90	65
7536545	274	34	23	60
7536545	271	12	11	30
6532510	373	11	47	45
6532510	372	60	66	88
6532510	371	23	45	67
5531519	433	63	70	90
5531519	421	40	66	96
5530535	433	38	60	100
5530535	421	50	80	77

```
insert into ogrenci
values(7536545,'Cengiz','Güneş',536,2,13,'04.11.1988',66,'Erkek')
```

Tablo 6.7: Ogrenci Tablosu

no	adi	soyadi	bolkod	sınıf	h_ID	d_tarihi	memleket	cinsiyet
5530535	Feyza	Uçan	530	4	7	1985-11-16	55	Bayan
5531519	Betül	Cebe	531	4	5	1986-06-01	5	Bayan
6532510	Leyla	Korkmaz	532	3	1	1986-09-11	1	Bayan
6532518	Ali	Yılmaz	532	3	1	1987-08-13	40	Erkek
6535501	Bilal	Baymaz	535	3	8	1987-07-21	28	Erkek
7536511	Aytül	Çetinkaya	536	2	13	1987-05-09	70	Bayan
7536521	Veli	Uşak	536	2	13	1988-07-03	21	Erkek
7536545	Cengiz	Güneş	536	2	13	1988-04-11	66	Bayan

Kaynaklar

- Veritabanı Yönetim Sistemleri, Turgut Özseven
- http://www.pau.edu.tr/ali/tr
- http://tr.wikipedia.org/wiki/Veritaban%C4%B1
- https://harunkolyigit.wordpress.com/2013/03/14/veritabani/
- http://akademik.maltepe.edu.tr/~senolerdogan/YZM_307/
- http://www.ortogon.com.tr/video/db/
- altanmesut.trakya.edu.tr/vt
- http://akademik.maltepe.edu.tr/~eminborandag/VeriTabani/