

# SQL DİLİ

VERİTABANI YÖNETİM SİSTEMLERİ

## SQL KATEGORİLERİ

İŞLEV	AÇIKLAMA
DDL (Data Definition Language – Veri Tanımlama Dili)	Veritabanı şemasını tanımlamak ve veritabanındaki veritabanı nesnelerinin yapısını oluşturmak ve değiştirmek için kullanılır.
DQL (Data Query Language - Veri Sorgu Dili)	Veritabanındaki verileri sorgulamak ve sonuçları listelemek için kullanılır.
DML (Data Manipulation Language - Veri İşleme Dili)	Veritabanında bulunan verilerin değiştirilmesi (güncellenmesi, eklenmesi, silinmesi) için kullanılır.
DCL (Data Control Language – Veri Kontrol Dili)	Grup ve kullanıcılara veritabanı sisteminde verilen izinleri ve konan yasakları düzenlemek amacıyla kullanılır.
TCL (Transaction Control Language – İşlem Kontrol Dili)	Veritabanındaki işlemlerle ilgili bütünlük vb. için kullanılır.

Kaynak: “İlişkisel Veritabanı Tasarlama ve SQL ile Geliştirme”, Dr. Selçuk Kıran, Nobel Akademik Yayıncılık

## SQL

SQL deyimleri yazılırken aşağıdaki kurallar göz önünde bulundurulmalıdır;

- SQL deyimleri büyük-küçük harf ayrimından etkilenmez.
- Bir SQL deyimi bir ya da birden fazla satırda yer alabilir.
- SQL anahtar kelimeleri kısaltılamaz ya da yarıda kesilerek ayrı satırlara bölünemez.

Kaynak: <http://www.cansuayvazguven.com>

## SQL VERİ TİPLERİ

## Karakter Tabanlı (Metinsel)

- **char:** Unicode olmayan sabit uzunlukta karakter verisi saklamak için kullanılır. Boyutu değişken olmakla birlikte en fazla 8000 karakter veri saklayabilir.
- **varchar:** Unicode olmayan değişken uzunlukta karakter verisi saklamak için kullanılır. Belirlenmiş veri kapasitesi 8000 karakter olmasına rağmen (max) parametresi ile bu değer  $2^{31}$  byte'a kadar genişletilebilir.
- **text:** Daha önceki sql versiyonlarını desteklemek amacıyla Sql Server 2012 içerisinde hala yer almaktadır. Bu veri tipi yerine varchar(max) kullanın.

Kaynak: <http://www.cansuayvazguven.com>

## Karakter Tabanlı (Metinsel)

- **nchar:** Sabit uzunlukta 4000 karakter unicode karakter verisi saklamak için kullanılır. Boyutu değişken olmakla birlikte kısa olan değerler atanan uzunluğa tamamlanır.
- **nvarchar:** Değişken uzunlukta unicode karakter verisi saklamak için kullanılır. Belirlenmiş maksimum uzunluk 4000 karakter olmasına rağmen (max) parametresiyle  $2^{31}$  byte'a kadar veri depolanabilir.
- **ntext:** text veri tipi gibi, bu veri tipide sadece eski versiyonları desteklemek amacıyla mevcuttur. Text tipinden tek farkı unicode veri saklayabilir.

Kaynak: <http://www.cansuayvazguven.com>

## Nümerik (Sayısal)

- **bit:** 1 byte uzunluğunda tamsayı veri tipidir. Boyutu aklınızı karıştırmayın. Genellikle evet/hayır şeklinde mantıksal bilgileri tutmak için kullanılır.
- **tinyint:** 1 byte büyüklüğünde 0 ile 255 arasında değer alabilen tamsayı veri tipidir.
- **smallint:** 2 byte büyüklüğünde -32768 ile 32767 arasında değer alabilen tamsayı veri tipidir.
- **int:** 4 byte büyüklüğünde yaklaşık -2 milyar ile +2 milyar arasında değer alabilen tamsayı veri tipidir.

Kaynak: <http://www.cansuayvazguven.com>

## Nümerik (Sayısal)

- **bigint:** 8 byte büyüklüğünde  $-2^{63}$  ile  $+2^{63}$  arasında değer alabilen tamsayı veri tipidir.
- **decimal ve numeric:** Decimal ve nümeric veri tipleri adları farklı olmasına rağmen kullanımları aynıdır. Boyutu değişken olmakla birlikte  $-10^{38}$  ile  $+10^{38}$  arasında ondalık ve tamsayı türünde verileri saklayabilir.
- **float:** Boyutu ve doğruluğu (ondalık kısım duyarlılığı) aldığı parametreye göre değişen kayan noktalı sayılar için kullanılır.
- **real:** Float veri tipinin parametre olarak 24 değerini aldığı ortaya çıkan halidir.  $-3.40E+38$  ile  $3.40E+38$  arasında değer alır.

Kaynak: <http://www.cansuayvazguven.com>

## Parasal

- **smallmoney:** 4 byte uzunluğunda yaklaşık -214 000 ile 214 000 arasında parasal değerleri tutmak için kullanılır. Ondalık kısım duyarlılığı 4 basamaktır.
- **money:** 8 byte uzunluğunda yaklaşık -922 milyar ile 922 milyar arasındaki parasal değerleri tutmak için kullanılır. Ondalık kısım duyarlılığı 4 basamaktır.

Kaynak: <http://www.cansuayvazguven.com>

## Tarih-Zaman

- **date:** Tarihleri YYYY-MM-DD şeklinde saklayan 3 byte uzunluğunda veri tipidir. 0001-01-01 ile 9999-12-31 tarihleri arasındaki tüm değerleri tutabilir.
- **smalldatetime:** Tarih ve zaman verilerini YYYY-MM-DD hh:mm:ss şeklinde saklayan 4 byte uzunluğunda veri tipidir. 1900-01-01 00:00:00 ile 2079-06-06 23:59:59 arasındaki tarih ve zaman değerlerini saklar.
- **datetime:** YYYY-MM-DD hh:mm:ss:mmm şeklinde tarih ve zaman verilerini tutan 8 byte uzunluğunda veri tipidir. 1753-01-01 00:00:00.000 ile 9999-12-31 23:59:59.999 arası değerlerini saklar.

Kaynak: <http://www.cansuayvazguven.com>

## Tarih-Zaman

- **datetime2:** datetime ile arasındaki fark salise hassasiyetinin daha yüksek olmasıdır. 7 basamağa kadar salise hassasiyeti kullanıcı tarafından belirlenebilir. Kapladığı alan salise hassasiyetine göre değişir. (6-8 byte)
- **time:** Sadece saat verilerini hh:mm:ss:nnnnnnn şeklinde saklayan veri tipidir. Datetime2 gibi salise hassasiyeti maksimum 7 basamaktır ve kullanıcı tarafından değiştirilebilir. (3-5 byte)
- **datetimeoffset:** Ükelere göre değişen zaman farkını tutmak için kullanılır

Kaynak: <http://www.cansuayvazguven.com>

## SQL OPERATÖRLER

## Karşılaştırma Operatörleri

OPERATÖR	ANLAMI
<	Küçük
>	Büyük
=	Eşit
<=	Küçük veya eşit
>=	Büyük veya eşit
	Eşit değil
!=	Eşit değil
LIKE	Metin Karşılaştırma Operatörü

## Mantıksal Operatörler

OPERATOR	ANLAMI
AND	Seçilecek satır için koşullardan en az biri doğru olmalıdır.
OR	Bir satırın seçilmesi için belirtilen tüm koşullar doğru olmalıdır.
NOT	Bir satırın seçilebilmesi için belirtilen koşulun yanlış olması gerekir.

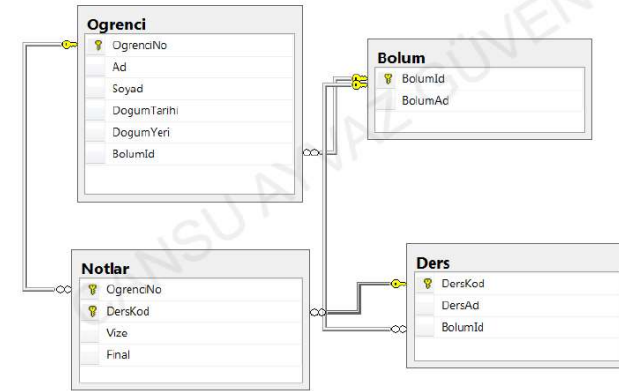
## Aritmetiksel Operatörler

OPERATÖR	İŞLEVİ
%	Mod alma
*	Çarpma
/	Bölme
+	Toplama
-	Çıkartma

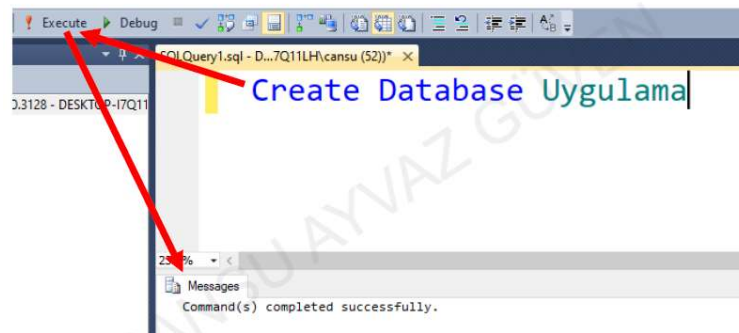
VERİ TANIMLAMA DİLİ (DDL)  
*CREATE – ALTER – DROP*

# CREATE

## Örnek Veritabanı



Kaynak: <http://www.cansuayvazguven.com>



Kaynak: <http://www.cansuayvazguven.com>

## Tablo Oluşturmak

Create Table *tablo\_ismi*( *sütun\_ismi1* veri\_tipi,  
*sütun\_ismi2* veri\_tipi,  
...)

Kaynak: <http://www.cansuayvazguven.com>

## Tablo Oluşturmak

```
Create Table ogrenci(  
  OgrNo int,  
  Ad varchar(25),  
  Soyad varchar(25),  
  DogumTarihi date)
```

Kaynak: <http://www.cansuayvazguven.com>

## Tablo Oluşturmak (Primary Key)

```
Create Table Bolum(  
  BolumId int not null Primary Key  
  identity (10,1),  
  BolumAd varchar(30))
```

Kaynak: <http://www.cansuayvazguven.com>

## Tablo Oluşturmak (Primary Key)

```
Create Table Ders(  
  DersKod int,  
  DersAd varchar(30),  
  BolumId int,  
  Constraint pkey Primary Key (DersKod))
```

Kaynak: <http://www.cansuayvazguven.com>

## Tablo Oluşturmak (Foreign Key)

```
Create Table Notlar(  
  OgrenciNo int,  
  DersKod int,  
  Vize int,  
  Final int,  
  Constraint fk_ono Foreign Key (OgrenciNo)  
    References ogrenci(OgrNo),  
  Constraint fk_dkod Foreign Key (DersKod)  
    References Ders(DersKod),  
  Constraint pkey Primary Key (OgrenciNo,DersKod))
```

Kaynak: <http://www.cansuayvazguven.com>



## Tablo Oluşturmak (Foreign Key)

```
Create Table ogrenci(  
  OgrNo int Primary Key ,  
  Ad varchar(25),  
  Soyad varchar(25),  
  DogumTarihi date,  
  BolumId int Foreign Key References Bolum(BolumId))
```

Kaynak: <http://www.cansuayvazguven.com>

## Tabloya Yeni Bir Sütun Eklemek

Alter Table *tablo\_ismi* Add *sütun\_ismi* özellikler

```
Alter Table ogrenci Add DogumYeri varchar(30)
```

Kaynak: <http://www.cansuayvazguven.com>

## Tabloda Belirli Sütunu Silmek

Alter Table *tablo\_ismi* Drop Column *sütun\_adı*

```
Alter Table ogrenci Drop Column DogumYeri
```

Kaynak: <http://www.cansuayvazguven.com>

## SQL KISITLAMALAR (CONSTRAINTS)



## NOT NULL

```
CREATE TABLE ogrenciler
(
  ad VARCHAR (50),
  soyad VARCHAR (50),
  bolum_ad VARCHAR (50) NOT NULL,
  bolum_kodu CHAR (4) NOT NULL,
  ogrenci_no INT NOT NULL
);
```

Örneğin ogrenciler adında bir tablo oluşturuyoruz. Bu tabloda öğrencilerin **bolum\_ad**, **bolum\_kodu**, **ogrenci\_no** alanlarına not null kısıtımızı ekliyoruz. Bu alanlar artık boş bırakılamayacaktır. Boş bırakıldığında hata verecektir.

Kaynak: <https://medium.com/@1gizemaltay/sql-constraints-kisitlemeler-ce1953057d1e>

## PRIMARY KEY

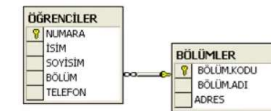
```
CREATE TABLE kişiler
(
  ID INT PRIMARY KEY NOT NULL,
  Ad VARCHAR (50) NOT NULL,
  Soyad VARCHAR (50)
);
```

Örneğin yukarıda oluşturulan kişiler tablosuna dikkat edilirse **ID** alanı tanımlanırken veri tipi belirtildikten sonra **PRIMARY KEY** kullanılmıştır. Bu durum, **ID** alanının birincil anahtar olduğunu ifade etmektedir.

Kaynak: <https://medium.com/@1gizemaltay/sql-constraints-kisitlemeler-ce1953057d1e>

## FOREIGN KEY

İki tablo arasında bağlantı kurmayı sağlayan referans anahtar veya dış anahtar adı verilen kısıtlamadır.



```
CREATE TABLE ogrenciler
(
  numara INT PRIMARY KEY NOT NULL,
  isim VARCHAR (25) NOT NULL,
  soyisim VARCHAR (25) NOT NULL,
  bolum CHAR (4) REFERENCES BÖLÜMLER(BÖLÜM_KODU)
  ON DELETE CASCADE ON UPDATE CASCADE NOT NULL,
  telefon CHAR (16) NULL
);
```

Kaynak: <https://medium.com/@1gizemaltay/sql-constraints-kisitlemeler-ce1953057d1e>



## UNIQUE

```
CREATE TABLE uyeler
(
  numara INT PRIMARY KEY NOT NULL,
  tc_no DECIMAL (11) UNIQUE NOT NULL,
  ad VARCHAR (50) NOT NULL,
  soyad VARCHAR (50) NOT NULL,
  telefon CHAR (16) NULL
);
```

Bu tabloda birincil anahtar olabilecek **numara** ve **tc\_no** adında iki alan bulunmaktadır. Bu alanlardan **numara** birincil anahtar olarak belirlenmiştir. **tc\_no** alanının ise tekrarlanması istenmemektedir. Bu nedenle **UNIQUE** kısıtı eklenmiştir. Böylelikle **tc\_no** birincil anahtar değildir ve tekrarlanmasına izin verilmeyecektir. Sadece bir defa girilmesine izin verilecektir.

Kaynak: <https://medium.com/@1gizemaltay/sql-constraints-kisitlemeler-ce1953057d1e>

## CHECK

Bir veritabanı tablosunda bulunan sütundaki tüm verilerin belirli bir koşulu karşılamasını sağlaması amacıyla kullanılır. Şartın sağlanıp sağlanmadığını kontrol eder. İstenmeyen durumların önlenmesi için tablolara check kısıtları eklenmektedir.

```
CREATE TABLE calisanlar
(
  numara INT PRIMARY KEY,
  ad VARCHAR (25) NOT NULL,
  soyad VARCHAR (25) NOT NULL,
  maaş SMALLMONEY NOT NULL,
  telefon CHAR (16) NULL,
  cinsiyet BIT NULL,
  CONSTRAINT CHK_maaş CHECK (maaş BETWEEN 1000 AND 2000)
);
```

Kaynak: <https://medium.com/@1gizemaltay/sql-constraints-kisitlemeler-ce1953057d1e>

## DEFAULT

Bir veritabanı tablosunda bazı alanların varsayılan bir değerinin olmasının istendiği durumlarda değer atamak için kullanılır. Alanların boş geçilmesini önlemek amacıyla değer atanır.

```
CREATE TABLE
(
  numara INT PRIMARY KEY,
  ad VARCHAR (25) NOT NULL,
  soyad VARCHAR (25) NOT NULL,
  bölüm CHAR (4) DEFAULT 'YBS' NOT NULL
);
```

Kaynak: <https://medium.com/@1gizemaltay/sql-constraints-kisitlemeler-ce1953057d1e>

## ALTER

## Tabloda Belirli Sütunun Özelliklerini Değiştirmek

Alter Table *tablo\_ismi* Alter Column *sütun\_adı*

```
Alter Table ogrenci Alter Column DogumTarihi smalldatetime
```

Kaynak: <http://www.cansuayvazguven.com>

## Tabloda Belirli Sütunun Özelliklerini Değiştirmek

```
Alter Table ogrenci Add Primary Key (OgrNo)
```

```
Alter Table ogrenci Add  
Constraint pk_ono Primary Key (OgrNo)
```

Var olan tablonun belirtilen sütununa Primary Key özelliği ekler.

Kaynak: <http://www.cansuayvazguven.com>

## Tabloda Belirli Sütunun Özelliklerini Değiştirmek

```
Alter Table Ders Add Foreign Key (BolumId)  
References Bolum(BolumId)
```

```
Alter Table Ders Add Constraint fk_bol  
Foreign Key (BolumId)  
References Bolum(BolumId)
```

Foreign Key özelliği ekler.

Kaynak: <http://www.cansuayvazguven.com>

## Tabloda Belirli Sütunun Özelliklerini Değiştirmek

Constraint Özelliğini Silmek  
Primary Key

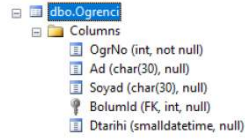
```
alter table ogrenci drop constraint pk_ono
```

dbo.Ogrenci  
Columns  
OgrNo (PK, int, not null)  
Ad (char(30), null)  
Soyad (char(30), null)  
BolumId (FK, int, null)  
Dtarihi (smalldatetime, null)

dbo.Ogrenci  
Columns  
OgrNo (int, not null)  
Ad (char(30), null)  
Soyad (char(30), null)  
BolumId (FK, int, null)  
Dtarihi (smalldatetime, null)

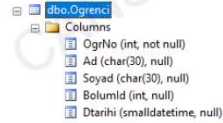
Kaynak: <http://www.cansuayvazguven.com>

## Tabloda Belirli Sütunun Özelliklerini Değiştirmek



dbo.Ogrenci
Columns
OgrNo (int, not null)
Ad (char(30), null)
Soyad (char(30), null)
BolumId (FK, int, null)
Dtarihi (smalldatetime, null)

```
alter table ogrenci drop constraint fk_bId
```



dbo.Ogrenci
Columns
OgrNo (int, not null)
Ad (char(30), null)
Soyad (char(30), null)
BolumId (int, null)
Dtarihi (smalldatetime, null)

Kaynak: <http://www.cansuayvazguven.com>

## DROP ve TRUNCATE

### Veritabanı Silmek

Drop Database *veritabanı\_ismi*

### Tablo Silmek

Drop Table *tablo\_ismi*

### Truncate

Tabloların içindeki tüm kayıtları siler.

Truncate Table *tablo\_ismi*

Kaynak: <http://www.cansuayvazguven.com>



<b>Yazar</b>	Selçuk Kıran
<b>Yayınevi</b>	Nobel Akademik Yayıncılık
<b>Barkod</b>	9786254069987
<b>ISBN</b>	978-625-406-998-7
<b>Sayfa Sayısı</b>	200 Sayfa
<b>Ürün Ebatı</b>	16x23,5
<b>Dil</b>	Türkçe
<b>Kağıt/Renk</b>	1. Hamur / Tek Renk
<b>Baskı Sayısı</b>	1. Baskı
<b>Baskı Yılı</b>	Ocak, 2021
<b>Kapak/Renk</b>	Karton Kapak