**İÇİNDEKİLER**

[VERİTABANI KOMUTLARI 2](#_Toc123631206)

[1. Veri Tanımlama Dili – Data Definition Language **(DDL)** 2](#_Toc123631207)

[1.1 Veritabanı Komutları 2](#_Toc123631208)

[1.2 Tablo Oluşturma Komutları 2](#_Toc123631209)

[1.3 Tabloda Değişiklik Komutları 3](#_Toc123631210)

[2. Veri İşleme Dili – Data Manipulation Language **(DML)** 3](#_Toc123631211)

[2.1 Sorgu Komutları 3](#_Toc123631212)

[2.2 Where Komut Kullanımı 3](#_Toc123631213)

[2.3 Sonuçların sıralanması 4](#_Toc123631214)

[2.4 Select Komutu Detayları 4](#_Toc123631215)

# VERİTABANI KOMUTLARI

* SQL = Structured Query Language - Yapısal Sorgulama Dili
* İlk başta matematiksel bir sözdizimine sahip olan SQUARE oluşturulmuş ardından geniş kitlelere hitap edebilmek adına İngilizce’ye benzer sözdizimine sahip SEQUEL(Structured English Query Language) oluşturulmuş, ardından SQL adını almıştır.
* ANSI ve ISO dili standartlaştırmak adına 1986da ANSI, 1987de ISO SQL standartlarını oluşturmuş.
* Komutlar temelde 3 alt grupta toplanır

|  |  |  |
| --- | --- | --- |
| SQL | | |
| Veri Tanımlama Dili – Data Definition Language (DDL) | Veri İşleme Dili – Data Manipulation Language **(DML)** | Veri Kontrol Dili – Data Control Language **(DCL)** |
| CREATE, ALTER, DROP | SELECT, INSERT, UPDATE, DELETE | GRANT, REVOKE, DENY |

# 1. Veri Tanımlama Dili – Data Definition Language **(DDL)**

* Veritabanının oluşturulması ve düzenlenmesi işlemlerini gerçekleştiririr. Veritabanı, Tablo, Trigger, View, Stored Prodecure nesneleri tanımalanabilir.

## 1.1 Veritabanı Komutları

Create komutu veri tabanı oluşturur. // CREATE DATABASE veritabani\_adi

Drop komutu veri tabanını içindeki kayıtlar ile birlikte siler. // DROP DATABASE veritabani\_adi

Alter komutu veritabanında güncelleme yapar;

ALTER DATABASE eskiAd MODIFY NAME = yeniAd  
ALTER DATABASE veritabaniAdi SET ONLİNE/OFFLİNE

## 1.2 Tablo Oluşturma Komutları

USE veritabani\_adi  
GO  
KOMUTLAR şeklinde veri tabanının yeri belirlenir

Tablo oluşturmak için;

CREATE TABLE tablo\_adi (  
alanAdi veriTipi (UK,PK,FK) (Null) (identity),  
alanAdi veriTipi,  
)

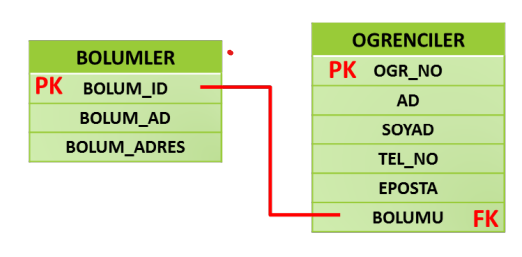
Tablo silmek için, içindeki tüm kayıtlar ile siler // DROP TABLE tablo\_adi

Constraint (Kısıt) oluşturmak(FK,PK,UK,IDENTITY,NULLABLE);

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Unique Key, Identity(1,1) kullanımı, // OGR\_No int UNIQUE, OGR\_Class int IDENTITY(1,1)  
IDENTITY(1,1) olduğunda 1den başlayarak 1er artar.

Foreign Key oluşturma yolları;  
-> BOLUMU int FOREIGN KEY(BOLUMU) REFERENCES BOLUMLER(BOLUM\_ID),  
-> CONSTRAINT FK\_Ogrenci\_Bolum FOREIGN KEY(BOLUMU) REFERENCES BOLUMLER(BOLUM\_ID)

Default için // Tel\_No varchar(11) DEFAULT ‘-’

## 1.3 Tabloda Değişiklik Komutları

Alan ekleme // ALTER TABLE tablo\_adi ADD alan\_adi veri\_tipi, …

Alan silme // ALTER TABLE tablo\_adi DROP COLUMN alan\_adi, …

Veri tipi değiştirme // ALTER TABLE tablo\_adi ALTER COLUMN alan\_adi veri\_tipi  
birden fazla yapmak için kodu tekrarlamak gerekir.

# 2. Veri İşleme Dili – Data Manipulation Language **(DML)**

* Veritabanında veriler üzerinde işlem yapar.

Tabloya veri eklemek;   
INSERT INTO tablo\_adi VALUES (değer1, değer2)  
INSERT INTO tablo\_adi (alan\_adi1, …) VALUES (değer1, …)

Tabloyu güncellemek;  
Tüm kayıtları güncellemek // UPDATE tablo\_adi SET alan1=değer1, alan2=değer2, …   
Belirli kayıtları güncellemek // UPDATE tablo\_adi SET alan1=değer1, alan2=değer2, … Where koşul

Tablodaki verileri silme;  
Tüm veriler silinecekse // DELETE FROM tablo\_adi  
Bazı kayıtlar silinecekse // DELETE FROM tablo\_adi WHERE koşul

## 2.1 Sorgu Komutları

Veritabanındaki kayıtları seçmek ve listelemek için kullanılır, parametreleri;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SELECT | FROM | WHERE | GROUP BY | HAVING | ORDER BY |
| Alan adı | Tablo adı | Koşullar | Gruplama Kriteri | Gruplama Koşulu | Sıralama Kriteri |

Kırmızı alanlar girilmesi zorunlu alanlardır, Sıralaması baştan sona doğrudur.

## 2.2 Where Komut Kullanımı

Filtreleme ve arama sorgularında Alan adı + Operatör + Değer şeklinde kullanılır.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| >, < | <=, =< | = | <> | NOT | LIKE | BETWEEN |
| Büyük, Küçük | Büyük Küçük eşit | Eşit | Farklı | Değil | Benzer | Arasında |

AND, OR, IN ifadeleri de koşul bağlaçlarıdır.

SELECT \* FROM ogrenci WHERE adi LIKE ‘F%’(F ile başlayan) veya ‘%oğlu’(oğlu ile biten) veya ‘%ay%’(ay içeren)  
SELECT ad FROM personel WHERE ad=’Ahmet’ AND dogumYeri=’İstanbul’

|  |  |
| --- | --- |
| METİNLER | TEK TIRNAK = ‘Ahmet’ |
| BİT ALANLAR | 1 veya 0 |
| TARİHLER | AA.GG.YYYY veya YYYY.AA.GG şeklinde |

Select ile seçilen verileri yeni tabloya kaydetme  
SELECT alan\_adlari INTO yeni\_tablo\_adi FROM tablo\_adi

## 2.3 Sonuçların sıralanması

ORDER BY ifadesi sıralama için kullanılır, varsayılan olarak ASC(ascending - artan sıralama) seçilidir. DESC(descending - azalan sıralama) da seçilebilir  
SELECT adi FROM ogrenci ORDER BY adi DESC

Distinct komutu tekrarlı verileri listelemeyi engeller, şehirler listesinde bir tane İstanbul göstermek gibi.  
SELECT DISTINCT dogum\_yeri FROM ogrenci

## 2.4 Select Komutu Detayları

Bazı sorguları takma ad ile listelemek istersiniz,  
SELECT ogr\_no AS Numara FROM ogrenci  
SELECT ogr\_no Numara FROM ogrenci  
SELECT Numara=ogr\_no FROM ogrenci

Year ve getdate komutu;  
YEAR() date tipindeki veriden sadece yılı alır, GETDATE() sistem tarihini verir.  
SELECT ogr\_no, adi, soyadi, 2021-YEAR(dogum\_tarihi) AS yaş FROM ogrenci  
SELECT ogr\_no, adi, soyadi, YEAR(GETDATE())-YEAR(dogum\_tarihi) AS yaş FROM ogrenci ORDER BY yaş

Metin alanlarını birleştirme concatenation;  
SELECT ogr\_ad + ‘ ‘ + ogr\_soyad AS ‘Ad Soyad’ FROM ogrenci ORDER BY ‘Ad Soyad’ DESC

Select sonucu dönen kayıt sayısını kısıtlama TOP komutu;  
SELECT TOP N alan\_adi FROM tablo\_adi (ilk N kayıt)  
SELECT TOP N PERCENT \* FROM tablo\_adi (ilk %N kayıt için)

NULL ile karşılaştırmak için SELECT \* FROM hocalar WHERE dogum\_tar IS NULL veya IS NOT NULL

|  |  |  |
| --- | --- | --- |
| VERİTABANI ve TABLO oluştururken kullanılan komutlar | *CREATE* | Oluştur |
| *DROP* | **Sil** |
| *ALTER* | **Düzenle** |
| Tabloya veriler eklerken kullanılan komutlar | *SELECT* | **Verileri Seç** |
| *INSERT* | **Veri Gir** |
| *UPDATE* | **Verileri Düzenle** |
| *DELETE* | **Verileri Sil** |