**SQL**

**Temel Komutlar**

* SELECT : Seçmek
* DELETE : Silmek
* UPDATE : Güncellemek
* INSERT : Kaydetmek

**Yerleşik SQL Fonksiyonları**

* AVG() : Kolonun ortalama değerini alır
* COUNT() : Satır sayısını verir
* ROUND() : Sayıları yuvarlar
* FIRST() : İlk değeri verir
* LAST() : Son değeri verir
* SUM() : Kolonun toplamını verir
* MAX() : En büyük değer
* MIN() : En küçük değer
* LEN() : Alanın karakter sayısını verir

**CREATE database**, veritabanının oluşturulmasını sağlar.

>> CREATE database sirket;

**Use**, ile veritabanları arasında geçiş yapabiliriz.

**CREATE table** ile veritabanı içerisinde tablo oluştururuz.

>> CREATE table Personeller (isim varchar(20),

                        soyisim varchar(20),

                        dogumtarihi date);

Tabloda bulunan bütün verileri görmek istersek **\*** işaretini kullanırız.

>> SELECT \* FROM Personeller;

Sadece tabloda bulunan belli bir değeri almak istiyorsak,

>> SELECT soyisim FROM Personeller;

Tablo içerisinde birden fazla aynı değer olabilir, benzersiz olacak şekilde bunları listelemek için **DISTINCT** kullanılır.

>> SELECT DISTINCT soyisim FROM Personeller;

Kriter, koşul ekmeke istersek eğer **WHERE** komutu kullanılır.

>> SELECT \* FROM Personeller WHERE dogumtarihi > 01/01/2000

**INSERT INTO** tabloya veri eklememizi sağlar

>> INSERT INTO Personeller (isim,soyisim,dogumtarihi)

                        VALUES ("Berkay","Yürür",29/03/1991);

**UPDATE** tablo içerisinde belirttiğimiz şarta uygun veriyi günceller.

>> UPDATE Personeller SET [soyisim] = "Kurhan" WHERE [isim] = "Ayşe"

**DELETE** şarta uygun veriyi tablodan siler.

>> DELETE FROM Personeller WHERE [isim] = "Ali"

**Allow Nulls**, işaretlenirse eğer boş satıra izin verilir.

Birden fazla, kolon adı ve komut kullanabiliriz

>> SELECT [isim],[soyisim] FROM Personeller WHERE [yas] > 22 AND [ulke] = "TR"

**LIKE** ile benzer değerleri listeleyebiliriz. Bunun için de **%** işareti kullanılır.

>> SELECT \* FROM Personeller WHERE [isim] LIKE "b%"

>> SELECT \* FROM Personeller WHERE [isim] LIKE "%b"

>> SELECT \* FROM Personeller WHERE [isim] LIKE "%b%"

**AS**, kolon başlığını değiştirmek istediğimiz zaman kullanılır.

**GROUP BY**, gruplandırma yapacağımız zaman kullanılırız. Hangi kolon için gruplandırma yapacaksak, en başta onu belirtmemiz gerekmektedir.

>> SELECT ulke, COUNT(ulke) AS "Ülke Sayısı" FROM Personeller GROUP BY ulke

**ALTER** **TABLE** ile seçili tabloda bulunan kolonlara ekleme, silme veya değiştirme yaparız.

>> ALTER TABLE Personeller ADD maas varchar(5);

>> ALTER TABLE Personeller DROP COLUMN maas;

**INNER** **JOIN**, iki tablo arasında eşleşen verileri birleştirir. Eşitlediğimiz yerler, kesişim noktalarıdır.

>> SELECT \* FROM Kategoriler INNER JOIN Urunler ON Kategoriler.kat\_id = Urunler.kad\_id;

**LEFT** **JOIN**, soldaki tablonun tamamını alır ve diğer tablodaki ilişkili verileri döndürür.

>> SELECT \* FROM Kitaplar LEFT JOIN Yazarlar ON Kitaplar.yazar\_id = Yazarlar.id;

**RIGHT** **JOIN**, sağdaki tablonun tamamını alır ve diğer tablodaki ilişkili verileri döndürür.

>> SELECT \* FROM Kitaplar RIGHT JOIN Yazarlar ON Kitaplar.yazar\_id = Yazarlar.id;

**FULL** **JOIN**, bütün tabloları alır, veri olmayan yerlere **NULL** yazar.

>> SELECT \* FROM Musteriler FULL JOIN Siparisler ON Musteriler.musteriler\_id = Siparisler.siparisler\_id

**COUNT**, toplam kaç satır olduğu bilgisini verir.

>> SELECT COUNT(\*) FROM Musteriler;

>> SELECT COUNT(yas) FROM Personeller

**SUM**, seçilen kolonun toplamını verir.

>> SELECT SUM(yas) FROM Personeller

**MAX** ve **MIN** seçili kolonda bulunan en büyük ve en küçük değeri verir.

>> SELECT MAX(yas) FROM Personeller

>> SELECT MIN(yas) FROM Personeller

**AVG**, kolonun ortalamasını gösterir.

>> SELECT AVG(musteriler\_id) FROM Musteriler;

Farklı fonksiyonları aynı sorgu içerisinde de kullanabiliriz.

>> SELECT ROUND(SUM(yas) / AVG(musteriler\_id)) FROM Musteriler;

**Python Üzerinden Örnekler (Sqlite)**

>> import sqlite3

>> con = sqlite3.connect("kütüphane.db")

>> cursor = con.cursor()

>> def tablo\_olustur():

>>    cursor.execute("CREATE TABLE IF NOT EXISTS personel(isim, soyisim, memleket)")

>>    con.commit()

>> def veri\_ekle\_1():

>>    cursor.execute("INSERT INTO personel VALUES('Berkay','Yürür','Istanbul)")

>>    con.commit()

>> def veri\_ekle\_2(isim, soyisim, memleket):

>>    cursor.execute("INSERT INTO personel VALUES(?,?,?)",(isim, soyisim, memleket))

>>    con.commit()

>> def veri\_al\_1():

>>    cursor.execute("SELECT \* FROM personel")

>>    liste = cursor.fetchall()

>>    for i in liste:

>>        print(i)

>> def veri\_al\_2(memleket):

>>    cursor.execute("SELECT \* FROM personel WHERE memleket = ?",(memleket,))

>>    liste = cursor.fetchall()

>>    for i in liste:

>>        print(i)

>> def veri\_guncelle(isim,memleket):

>>   cursor.execute("UPDATE personel SET memleket = ? WHERE isim =?",(memleket,isim,))

>>    con.commit()

>> def veri\_sil(isim):

>>    cursor.execute("DELETE \* FROM personel WHERE isim = ?",(isim,))

>>    con.commit()

>> tablo\_olustur()

>> isim = input("İsim  : ")

>> soyisim = input("Soyisim  : ")

>> memleket = input("Memleket  : ")

>> veri\_ekle\_2(isim,soyisim,memleket)

>> veri\_al\_1()

>> memleket\_2 = input("Listelenecek il bilgisini giriniz : ")

>> veri\_al\_2(memleket\_2)

>> con.close()