create table urun(

    id integer primary key not null,

    ad varchar(15) not null,

    marka varchar(20),

    stok integer,

    kategori varchar(15)

)

* Create table table\_name(

Columns\_name columns\_type columns\_defination,

)

* + Tablo oluşturma syntax
    - Create🡪 oluşturma komutu
    - Table🡪 tip bilgisi
    - Columns\_type🡪 integer/char gibi tipleri
    - Columns\_defination🡪 primary key not null, not null gibi özel belirteçler
    - Ekstra\*\* -> eğer char/varchar() gibi değişkenler kullanıyorsak parantez içerisine maksimum girebileceği harf sayısını yazabiliriz ‘varchar(20)’

drop table urun

* Drop table table\_name
  + Tablo silme syntaax
    - Drop🡪 silme komutu
    - Table🡪tip bilgisi
    - Table\_name🡪 silinecek tablonun adı

insert into musteri(id,name,surname,city) values (8,'Tufan','Çimen','İstanbul')

* İnsert into table\_name(table\_columns []) values (columns new values[])
  + Tabloya veri ekleme syntax
    - İnsert –-> ekleme komutu
    - İnto 🡪 nerenin içerisine ekleyeceğini gösteren belirteç
    - Table\_name 🡪 eklenecek olan tablonun adı
    - Table\_columns[] 🡪 eklenecek olan tablodaki columns lar
    - Columns new values[] 🡪 columnsların karşılığına denk gelecek olan veriler

select id,name,surname from musteri

* Select [neye göre seçeceği] from [nerden çekeceği]

select city from musteri

* Select [tekil bir kategoriye göre] from [nereden çekeceği]

select \* from musteri

* Select \*(hepsini al bildirgeçi) from [nereden çekeceği]

select \* from musteri where name='Mehmet'

* Select \* from table\_name where [neye göre ayıklayacağı]

select \* from musteri where city='Ankara' and id = 4 or city='Adana'

* Select \* from table\_name where [neye göre ayıklayacağı]

select\* from musteri where bakiye>3000

* Select \* from table\_name where [neye göre ayıklayacağı]

select \* from musteri where name like 'As%'

* Select \* from table\_name where [isminde baştan As içeren (like) verileri ver),

select\*from musteri order by id

* Select \* from table\_name order by \_columns\_name

select \* from musteri where name not like '%a%'

* Select \* from table\_name where [isminde a içermeyen (not like) verileri göster)
  + Tabloya veri seçme/gösterme/aratma syntax
    - Select 🡪 gösterme/seçme komutu
    - \* 🡪 hepsini al
    - From 🡪 nerden geldiğini anlayan bildirgeç
    - Table\_name 🡪 tablonun adı
    - Where 🡪 neye göre sıralayacağı
    - Name🡪 tablodaki verinin columns adı
    - Like 🡪 içerenleri bulur
    - Not like🡪 içermeyenleri bulur
    - Bakiye > 3000 🡪 3000 den yüksek bakiyeleri gösterir
    - ‘As%’ 🡪 % işareti sonda olduğu için As ile başlayan anlamına gelir
    - ‘%a%’ 🡪 her iki tarafta olduğu için direk a içerenleri bulur
    - Order by 🡪 tabloyu sıralamak için kullanıyoruz

DELETE from musteri where id=10

* Delete from table\_name where \_columnsname=columns\_location
  + Tablodan veri silme syntax
    - Delete🡪 silme komutu
    - From 🡪 yer bildirgeci
    - Table\_name 🡪 verinin silineceği tablonun adı
    - Where 🡪 neye göre sileceğinin bildireci

update musteri set bakiye=5750

* Update table\_name set columns\_name=value
  + Update veri güncelleme syntax
    - Update 🡪 güncelleme komutu
    - Table\_name 🡪 erişelecek tablonun adı
    - Set 🡪 sütunu set etme komutu
    - Columns\_name🡪 değiştirilecek sütunun adı
    - Value 🡪 yeni atanacak değer

update musteri set bakiye=5750 where id=1

* Update table\_name set columns\_name=value where columns\_name=value
  + Tablodaki seçilen bir hedefin değerini değiştirmek
    - Where 🡪 tablodaki sütundaki hedefi bulma komutu
    - Columns\_name🡪 hedef alınan sütunun adı
    - Value🡪 yeni atanacak değer

select count(\*) from musteri

select count(\*) from musteri where city='Ankara'

select count(\*) from musteri where city='Ankara' and bakiye>5000

select count(\*) from musteri where city='Ankara' or bakiye>5000

* Select count(\* or columns\_name) from table\_name where columns\_name=value and/or columns\_name >/</= value
  + Count operetörü syntax, count operatörü miktarı ifade eder
    - Count() 🡪 miktar methodu

select sum(bakiye) from musteri

select sum(bakiye) from musteri where city='Ankara'

select sum(bakiye) from musteri where city!='Ankara'

* Select sum(columns\_name) from table\_name where columns\_name =/!= value
  + Sum operatörü syntax, sum operatörü toplam orda bulunan sayıyı ifade eder

select avg(bakiye) from musteri

select avg(bakiye) from musteri where city='Ankara'

* Select avg(columns\_name) from table\_name where columns\_name=value
  + Avg operatörü syntax, avg operatörü columnsta bulunan verilerin ortalamasını verir

select max(bakiye) from musteri

select min(bakiye) from musteri

select min(bakiye),max(bakiye) from musteri

select max(bakiye)-min(bakiye) from musteri

select max(bakiye) from musteri where city='Bursa'

* Select max(column\_name)/min(columns\_name) from musteri where columns\_name=value
  + Max() 🡪 columns daki en yüksek değeri verir
  + Min() 🡪 columns daki en düşük değeri verir

select city,count(\*) from musteri group by city

select city,count(\*) as kişi from musteri group by city

select city,count(\*) as kişi from musteri group by city order by count(\*)

select city,count(\*) as kişi from musteri group by city order by count(\*) desc

* Select columns\_name,count(\*) as kişi from table\_name group by columns\_name order by count(\*) desc
  + Group by 🡪 bir grubu oluşturup o grup için iç içe sorgu yapmamıza olanak sağlıyor

select city, sum(bakiye) from musteri group by city

select city,avg(bakiye) from musteri group by city

* Select columns\_name,sum(columns\_name)/avg(columns\_name) from table\_name group by columns\_name
  + Buradaki group by ile city’i verilerini alıyoruz ve o verileri cityden toplarını veya ortalamalarını bakıyoruz

select city,count(\*) from musteri group by city  having count(\*)>3

* Select columns\_name,count(\*) from musteri group by columns\_name having count(\*)>value
  + Burada yine aynı şekilde city miktarlarını alıyoruz ve miktarı 3 den fazla olanları çekiyoruz.

select avg(bakiye) as ortalama,city from musteri group by city

select avg(bakiye) as ortalama,city from musteri group by city having avg(bakiye)>5000 or city like '%u%'

select \* from musteri where bakiye=(select max(bakiye) from musteri where sehir='İstanbul')

* Select \* from table\_name where columns\_name=(select max(columns\_name) from table\_name where columns\_name2=value)
  + Alt sorgu sayesinde iç içe sorgulama yapıp istediğimiz veriye ulaşmaya çalışıyoruz
  + Burda amaç istanbul şehrinde bakiyesi en yüksek olanları sadece görmek.

select \* from musteri where meslek=(select id from meslek where ad='mühendis') order by id

* Select \* from table\_name where columns\_name=(select columns2\_name from table2\_name where columns2\_name=value) order by columns\_name
  + Alt sorguları kullanarak farklı 2 tablonun verilerini eşitliyoruz.
  + Burda amaç musteri tablosunda id olarak atanan meslekleri meslek tablosuna eşitleyip stringlerini görmek

update musteri set bakiye=bakiye+bakiye\*0.1 where meslek=(select id from meslek where ad ='mühendis')

* Update table\_name set columns\_name=formulas where columns\_name=(select columns\_name2 from table2\_name where columns\_name=value)
  + Alt sorguları kullanarak meslek grubuna özel bakiye artışı sağlıyoruz
  + Burda musteri tablosunda müsterilerin bakiyesini belli bir meslek grubuna özel artış sağlıyoruz, bunu sağlarken eklenen diğer tablo ile denkleştirip bakiye artışını sağlayabiliriz.

select \* from musteri where sehir='Ankara' and meslek=(select id from meslek where ad='öğretmen')

* Select \* from table\_name where columns\_name=value and columns\_name=(select columns2\_name from table\_name where columns\_name=value)
  + Alt sorguları kullanarak ankara da bulunan öğretmenlerin verisin elde etmeye çalışıyoruz
  + Ankarada bulunan öğretmenlere erişmek istiyoruz bunu yaparken musteri tablosunda mesleklerin idlerini meslek tablosuna ilişkilendirerek ankarada bulunan öğretmenlerin verisini çekeriz.

truncate table urun

* Truncate table table\_name
  + Truncate tabloda bulunan verileri temizler ve temiz bir tablo ile devam ederiz.

select musteri.ad,soyad,sehir,meslek.ad from musteri inner join meslek on musteri.meslek=meslek.id

* Select table\_name.columns\_name, columns\_name,columns\_name, table2\_name.columns2\_name from table\_name inner join table2\_name on table\_name.columns\_name=table2\_name.columns\_name
  + Musteri tablosunda id olarak atanan meslekleri meslek tablosuna bağlayıp ilişki veri çekiyoruz
  + Bunu yaparken önce istediğimiz verilerin columnsları çekiyoruz daha sonra diğer tablonun mesleklerini ilk tablodaki id lerin bulunduğu kısma eşitliyoruz.

select bolumid,bolumad,ad from bolum left join fakulte on bolum.bolumf=fakulte.id

select id,ad,bolumad from fakulte right join bolum on fakulte.id=bolum.bolumf

* Select columns\_name,columns\_name,columns\_name from table\_name left join fakulte on bolum.bolumf=fakulte.id
  + Bolum tablosunu fakulte tablosuna bağlıyoruz fakat left baglacı ile soldakini sağa bağlıyor aynı komutun right halide sağdakine bağlıyor

select musteri.name,musteri.surname,city,meslek.job from musteri full join meslek on musteri.job=meslek.id

* Sadece istenen kısmı sağı yada solu değil verilen değerlere göre iki tabloyu tamamen bağlar

select bolumad,ad from bolum cross join fakulte

* İki tabloyu bir birine eşitler her veriyi karşıdaki bir veriyle teker teker eşleştirir (1,1) (1,2) vs.

select \* from bolum2 intersect select \* from bolum3

* İki tablodaki ortak verileri verir

select \* from bolum2 except select \* from bolum3

* İki tablonun arasındaki farklı olan verileri gösterir

select \* from bolum2 union select \* from bolum3 order by id

* İki tabloyu birleştiriyor fakat aynı olanları tek olarak yazıyor

select \* from bolum2 union all select \* from bolum3 order by id

* İki tabloyu birleştiriyor fakat aynı olanları ayrı ayrı yazıyor

select ASCII ('A')

* Girilen karakterin ascii değerini veriyor

select concat('Günaydın ',' Sql',' Dersleri')

* İçeriği birleştiriyor ve içeriği bastırıyor

select concat\_WS('\*','Günaydın ',' Sql',' Dersleri')

* İlk parametreyi diğer parametrelerin arasına yerleştirip bastırır

select left ('Merhaba Dünya',3)

* Soldan 3 karakteri alır

select right ('Merhaba Dünya',3)

* Sağdan 3 karakteri alır

select length('benim manevi mirasım bilim ve akıldır')

* Kaç karakterden oluştuğunu söyler

select id,replace(adi,'i','ı') from bolum3

* ‘adi’ sütunu içinde bulunan i harflerini ı harfiyle değiştirir

select reverse(adi) from bolum3

* ‘adi’ sütununda bulunan verileri tersten yazdırır

select substring('Murat Yücedağ',4,3)

* Verilen verinin 4. Ve 3. Karakterini bastırır

select lower(adi) from bolum3

* ‘adi’ sütunundaki bütün verileri küçük harfe dönüştürür

select upper(adi) from bolum3

* ‘adi’ sütunundaki bütün verileri büyük harfli yapar

select abs(-5)

* Mutlak değer

 select ceil(4.7)/ceiling(4.7)

* Yukarı yuvarlama

select floor(4.7)

* Aşağıya yuvarlama

select pi()

* Pi sayısı

select power(2,5)

* Üstel sayılar

select Random()/Rand()

* Random sayı komutu

select Round(18.1234,2)

* İlk verilen verinen 2. Değerdeki kadar sayısını gösterir 18.12

select sign(25)

* İçine girilen verinin pozitif mi negatif mi olduğunu belirtir
* Pozitif 1, negatif -1, sıfır 0

select Sqrt(625)

* Karekök

select square(2)

* Karesini alır

select log(50)

* Logaritmik değerini verir

select current\_date

* Tarihi verir

select current\_time()

* mevcutta olan saati verir eğer parantez eklenip içine değer verirsek o değer kadar virgülden sonra salisei gösteri

select now()

* yerel saati verir

select age(timestamp '01.09.2020')

* girilen tarih ile şuanki tarihi karşılaştırır ve aradaki farkı söyler

select ad,tarih,age(now(),tarih) from kitaplar

* tabloda bulunan verilerin kitapların şuanki zamandan farkını yani ne kadar süre geçtiğini söyler