What is MySQL?

MySQL is a **relational database management system** (RDBMS) developed by Oracle that is based on structured query language (SQL). A database is a structured collection of data.

* MySQL is a database system used on the web.
* MySQL is a database system that runs on a server.
* MySQL is ideal for both small and large applications.
* MySQL is very fast, reliable, and easy to use.
* MySQL uses standard SQL.
* MySQL compiles on several platforms.
* MySQL is free to download and use.
* MySQL is developed, distributed, and supported by Oracle Corporation.

SQL is a standard language for storing, manipulating, and retrieving data in databases.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| musterino | ad | soyad | dtarih | sehir | cinsiyet | puan |
| 1 | Ahmet | Cansever | 1956-02-19 00:00:00.000 | İstanbul | E | 64 |
| 2 | Mehmet | Aydın | 1976-02-19 00:00:00.000 | Samsun | E | 55 |
| 3 | Aliye | Seven | 1966-06-10 00:00:00.000 | Konya | K | 45 |
| 4 | Burak | Sayın | 1996-02-19 00:00:00.000 | İstanbul | E | 23 |
| 5 | Beyza | Kılıç | 1955-12-30 00:00:00.000 | Manisa | K | 85 |

**SQL SELECT**

musteri tablosunda bulunan ad, soyad sütunlarını listelemek için;

|  |  |
| --- | --- |
| 1  2  3 | Select ad,soyad FROM musteri |

musteri tablosunda bulunan tüm kayıtları listelemek için;

|  |  |
| --- | --- |
| 1  2  3 | Select \*From musteri |

SQL SELECT DISTINCT

Bir tabloda bir sütun yinelenen değerleri içerebilir. Distinct ile farklı değerleri listeleyebiliriz.

|  |  |
| --- | --- |
| 1  2  3 | SELECT DISTINCT sehir FROM musteri; |

SQL WHERE

Where anahtar sözcüğü ile sadece belirlenen kurala uygun olan kayıtların listelenmesini sağlayabiliriz.

Örneğin musteri tablosunda sehir sütunu İstanbul olan kayıtları listelemek için;

|  |  |
| --- | --- |
| 1  2  3 | SELECT \* FROM musteri WHERE sehir='istanbul' |

yada cinsiyet “K” olan kayıtları listelemek için;

|  |  |
| --- | --- |
| 1  2  3 | SELECT \* FROM musteri WHERE cinsiyet='K' |

Where ile kullanabileceğimiz operatörler;

|  |  |
| --- | --- |
| **Operator** | **Açıklama** |
| = | Eşit |
| <> | EşitDeğil. **Note:** Bazı versiyonlarda “!=” kullanılabilir. |
| > | Büyüktür |
| < | Küçüktür. |
| >= | Büyük Eşit |
| <= | Küçük Eşit |
| BETWEEN | arasında |
| LIKE | Örüntü arama |
| IN | Bir sütun için birden çok olası değerleri belirtmek için |

SQL AND – OR Kullanımı

AND Operatörü 1. Koşul ve 2. Koşulun doğru olması durumunda çalışır.

Örneğin musteri tablosunda  Cinsiyeti “E” ve Şehri “İstanbul” olanları listelemek için;

|  |  |
| --- | --- |
| 1  2  3  4  5 | SELECT \* FROM musteri  WHERE sehir='İstanbul'  AND cinsiyet='E' |

OR operatörü 1. Koşul yada 2. Koşulun doğru olması durumunda çalışır.

Örneğin musteri tablosunda şehri İstanbul yada Samsun olanları listelemek için;

|  |  |
| --- | --- |
| 1  2  3  4  5 | SELECT \* FROM musteri  WHERE sehir='İstanbul'  OR sehir='Samsun' |

AND ve OR operatörü birlikte de kullanılabilir.

Örnek olarak musteri tablosunda cinsiyeti ‘K’ olan ve Şehri “Konya” yada “Manisa” olanları listelemek için.

|  |  |
| --- | --- |
| 1  2  3  4  5 | SELECT \* FROM musteri  WHERE cinsiyet='K'  AND (City='Konya' OR City='Manisa') |

SQL ORDER BY Kullanımı

ORDER BY, varsayılan olarak artan düzende kayıtları sıralar. Azalan kayıtları sıralamak için DESC anahtar sözcüğünü kullanabilirsiniz.

Örnek olarak musteri tablosundaki kayıtları ad sütünuna göre artan ve azalan olarak sıralayalım.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteri  ORDER BY ad |

Azalan sıralama örneği;

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteri  ORDER BY ad DESC |

INSERT INTO Kullanımı

Kayıt eklemek için kullanılır.

örnek olarak musteri tablosuna bir kayıt ekleyelim.

|  |  |
| --- | --- |
| 1  2  3  4 | INSERT INTO musteri(ad, soyad, dtarih, sehir, cinsiyet, puan)  VALUES ('Ali','Şahin','2000-10-12','Burdur','E',68) |

SQL UPDATE kullanımı

Kayıtlar üzerinde değişiklik güncelleme yapmak için kullanılır.

Örnek olarak musterino su 3 olan kaydın puanını 90 olarak değiştirelim.

|  |  |
| --- | --- |
| 1  2  3  4  5 | UPDATE musteriler  SET puan=90  WHERE musterino=3 |

SQL DELETE Kullanımı

Tablodan kayıt silmek için kullanılır.

Örnek olarak musterino su 4 olan kaydı silmek için

|  |  |
| --- | --- |
| 1  2  3  4 | DELETE FROM musteriler  WHERE musterino=4 |

musteriler tablosundaki tüm kayıtları silmek için

|  |  |
| --- | --- |
| 1  2  3 | DELETE \* FROM musteriler |

SQL SELECT TOP Kullanımı

Belirtilen sayıda kaydı görüntülemek için kullanılır.  
örnek olarak musteriler tablosundaki ilk 5 kaydı listeleylim.

|  |  |
| --- | --- |
| 1  2  3 | SELECT TOP 5 \* FROM musteriler |

SQL LIKE Kullanımı

Belirtilen bir değeri aramak için kullanılır.  
Örnek olarak musteriler tablosunda şehri S ile başlayan kayıtları listeleyelim.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE sehir LIKE 's%' |

Şehri s ile biten kayıtları listelemek için;

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE sehir LIKE '%s' |

Şehrin içerisinde “tan” bulunan kayıtları listelemek için;

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE sehir LIKE '%tan%' |

SQL YER TUTUCU KARAKTERLER

musteriler tablosunda adı “al” ile başlayan kayıtları listelemek için;

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE AD LIKE 'al%' |

musteriler tablosunda adı “al” ile biten kayıtları listelemek için;

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE AD LIKE '%al' |

musteriler tablosunda ismi A ile başlayıp ondan sonraki 2 karakteri herhangi bir harf olan ve e ile devam eden ve ondan sonraki harfi belli olmayan kayıtları listeleyelim.(Adı Ahmet olanları listeleyeceğiz. :))

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE ad LIKE 'A \_ \_ e \_' |

Adı a ile b ile ya da s ile başlayan kayıtları listeleyelim.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE adLIKE '[abs]%' |

şimdide tam tersi a ile b ile yada s ile **başlamayan**kayıtları listeleyelim.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE adLIKE '[!abs]%' |

SQL IN Kullanımı

IN operatörü, WHERE yan tümcesinde birden fazla değer belirlemenizi sağlar.  
Örnek olarak şehri İstanbul ve Konya olan kayıtları listeleyelim.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE sehir IN ('İstanbul','Konya') |

SQL BETWEEN KULLANIMI

Between operatörü belirli kriterler arasındaki kayıtları listelemek için kullanılır. Sayı metin yada tarih aralığı verilebilir.

Örnek olarak musteriler tablosunda puanı 70 ile 90 arasında olan kayıtları listeleyelim.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE puanNOT BETWEEN 70 AND 90 |

Doğum tarihi 01/01/1996 ile 01/01/2006 arasındaki kişileri listelemek için;

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE dtarih BETWEEN '01/01/1996' AND '01/01/2006' |

musteriler tablosunda adı C ve E arasında olan kayıtları listeleyelim.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \* FROM musteriler  WHERE ad  BETWEEN 'C' AND 'E' |

SQL ALIASES Kullanımı

SQL tablosunda yer alan sütunlara geçici bir ad vermek için kullanılır.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT ad AS ADI, soyad AS SOYADI,dtarih AS [DOĞUM TARİHİ]  FROM musteriler |

SQL JOIN Kullanımı

Orders ve Customers tablolarını kullanarak oluşturulan örneği inceleyelim.

|  |  |  |
| --- | --- | --- |
| **OrderID** | **CustomerID** | **OrderDate** |
| 10308 | 2 | 1996-09-18 |
| 10309 | 37 | 1996-09-19 |
| 10310 | 77 | 1996-09-20 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CustomerID** | | **CustomerName** | **ContactName** | **Country** | |
| 1 | | Alfreds Futterkiste | Maria Anders | Germany | |
| 2 | | Ana Trujillo Emparedados y helados | Ana Trujillo | Mexico | |
| 3 | | Antonio Moreno Taquería | Antonio Moreno | Mexico | |
| 1  2  3  4  5  6 | SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  FROM Orders  INNER JOIN Customers  ON Orders.CustomerID=Customers.CustomerID; | | | |

Sorgumuzu çalıştırdığımızda

|  |  |  |
| --- | --- | --- |
| **OrderID** | **CustomerName** | **OrderDate** |
| 10308 | Ana Trujillo Emparedados y helados | 9/18/1996 |
| 10365 | Antonio Moreno Taquería | 11/27/1996 |
| 10383 | Around the Horn | 12/16/1996 |
| 10355 | Around the Horn | 11/15/1996 |
| 10278 | Berglunds snabbköp | 8/12/1996 |

SQL LEFT JOIN Kullanımı

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | SELECT Customers.CustomerName, Orders.OrderID  FROM Customers  LEFT JOIN Orders  ON Customers.CustomerID=Orders.CustomerID  ORDER BY Customers.CustomerName |

SQL RIGHT JOIN Kullanımı

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | SELECT Orders.OrderID, Employees.FirstName  FROM Orders  RIGHT JOIN Employees  ON Orders.EmployeeID=Employees.EmployeeID  ORDER BY Orders.OrderID |

SQL SELECT INTO Kullanımı

SELECT INTO deyimi bir tablodan veri seçer ve yeni bir tabloya ekler.

örnek olarak musteriler tablosunun musterilerbackup isimli yedeğini oluşturabiliriz.

|  |  |
| --- | --- |
| 1  2  3  4  5 | SELECT \*  INTO musterilerbackup  FROM musteriler |

Sadece şehri “İstanbul” olan kayıtları eklemek için;

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | SELECT \*  INTO musterilerbackup  FROM musteriler  WHERE sehir='İstanbul' |

SQL CREATE Kullanımı

Yeni bir database oluşturmak için;

|  |  |
| --- | --- |
| 1  2  3 | SQL CREATE DATABASE dbname |

SQL CREATE TABLE Kullanımı

Yeni tablo oluşturmak için kullanılır.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | CREATE TABLE musteriler  (  id int,  ad varchar(255),  soyad varchar(255),  adres varchar(255),  sehir varchar(255)  ); |

SQL AVG  Kullanımı

AVG () işlevi sayısal bir sütunun ortalama değerini döndürür.

Örnek olarak musteriler tablosunda puan ortalamasını bulalım.

|  |  |
| --- | --- |
| 1  2  3 | SELECT AVG(puan) FROM musteriler |

Aşağıdaki sorguda ise musteriler tablosunda puanı, puan ortalamasının üstünde yer alan kayıtlara ait ad ve soyad verileri listelenmektedir.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT ad, soyad FROM musteriler  WHERE puan&gt;(SELECT AVG(puan) FROM musteriler); |

SQL COUNT Kullanımı

COUNT () işlevi, belirtilen ölçütlerle eşleşen satır sayısını döndürür.

Örnek olarak musteriler tablosundaki toplam kayıt sayısını bulalım.

|  |  |
| --- | --- |
| 1  2  3 | SELECT COUNT(\*) FROM musteriler; |

Şimdide musteriler tablosunda kaç farklı şehir olduğunu bulalım.

|  |  |
| --- | --- |
| 1  2  3 | SELECT COUNT(DISTINCT sehir) FROM musteriler; |

Şimdi de Şubat ayında doğan Müşterileri sayısını bulalım.

|  |  |
| --- | --- |
| 1  2  3 | SELECT COUNT (\*) FROM musteriler  WHERE MONTH(dtarih)=2 |

Aşağıdaki SQL Deyimi ise “Orders” tablosundan “CustomerID” = 7 ye ait siparişlerin sayısını bulur.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT COUNT(CustomerID) AS OrdersFromCustomerID7 FROM Orders  WHERE CustomerID=7 |

SQL ROUND Kullanımı

Ondalıklı sayıyı yuvarlamak için kullanılır.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT urunadi, ROUND(fiyat,0) AS YuvarlanmisFiyat  FROM urunler |

SQL LEN () KULLANIMI

LEN () işlevi, bir metin alanındaki değerin uzunluğunu döndürür.  
Örnek olarak musteriler tablosunda ad alanını yanına soyad uzunluklarını yazdıralım.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT ad,LEN(soyad) as soyaduzunlugu  FROM musteriler; |

SQL NOW () Kullanımı

NOW fonksiyonu sistem tarih ve saatini döndürür.

Örnek olarak Aşağıdaki SQL deyimi “Products” tablosundan bugün için ürün adını ve fiyatını seçelim.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT ProductName, Price, Now() AS PerDate  FROM Products; |

SQL MAX Kullanımı

MAX () işlevi, seçilen sütunun en büyük değeri döndürür.

Örnek olarak musteriler tablosunda en yüksek puanı görüntüleyelim.

|  |  |
| --- | --- |
| 1  2  3 | Select MAX(puan) AS EnyuksekPuan FROM musteriler |

Şimdide en yüksek puana sahip müşterileri ad ve soyadı ile listeleyelim.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | SELECT ad,soyad,puan  FROM musteriler  WHERE puan =(SELECT MAX(puan)  FROM musteriler) |

SQL MIN Kullanımı

MIN () işlevi, seçilen sütunun en küçük değeri döndürür.

Ogrenciler tablosunda en düşük puanı görüntüleyelim.

|  |  |
| --- | --- |
| 1  2  3 | Select MIN(puan) AS EndusukPuan FROM ogrenciler |

Şimdide en düşük puana sahip ogrencileri ad ve soyadı ile listeleyelim.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | SELECT ad,soyad,puan  FROM ogrenciler  WHERE puan =(SELECT MAX(puan)  FROM ogrenciler) |