

* SELECT \* FROM Customers;
* SELECT - extracts data from a database
* UPDATE - updates data in a database
* DELETE - deletes data from a database
* INSERT INTO - inserts new data into a database
* CREATE DATABASE - creates a new database
* ALTER DATABASE - modifies a database
* CREATE TABLE - creates a new table
* ALTER TABLE - modifies a table
* DROP TABLE - deletes a table
* CREATE INDEX - creates an index (search key)
* DROP INDEX - deletes an index
* SELECT CustomerName,City FROM Customers;
* SELECT DISTINCT Country FROM Customers;
* SELECT COUNT(DISTINCT Country) FROM Customers;

yukaridaki komut firefoxta calismaz.

* SELECT Count(\*) AS DistinctCountries

FROM (SELECT DISTINCT Country FROM Customers);

bu calisir iste!

* SELECT \* FROM Customers

WHERE Country='Mexico';

* SELECT \* FROM Customers

WHERE Country='Germany' AND City='Berlin';

* SELECT \* FROM Customers

WHERE NOT Country='Germany';

* SELECT \* FROM Customers

WHERE Country='Germany' AND (City='Berlin' OR City='München');

* SELECT \* FROM Customers

ORDER BY Country;

* SELECT \* FROM Customers

ORDER BY Country DESC;

ya da ASC de kullanilabilirdi!

* SELECT \* FROM Customers

ORDER BY Country, CustomerName;

The following SQL statement selects all customers from the "Customers" table, sorted by the "Country" and the "CustomerName" column. This means that it orders by Country, but if some rows have the same Country, it orders them by CustomerName!!!

* SELECT \* FROM Customers

ORDER BY Country ASC, CustomerName DESC;

* INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('Cardinal','Tom B. Erichsen','Skagen 21','Stavanger', '4006','Norway');

eger yeni bir sutun eklenecekse boyle yapilir ama eger var olan sutunlara bir ekleme yapilacaksa asagidaki gibi yapilir!!!

* INSERT INTO Customers

VALUES ('Cardinal', 'Stavanger', 'Norway');

* SELECT CustomerName, ContactName, Address

FROM Customers

WHERE Address IS NULL;

* UPDATE Customers

SET ContactName='Alfred Schmidt', City='Frankfurt'

WHERE CustomerID=1;

* DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
* SELECT TOP 3 \* FROM Customers;
* SELECT TOP 50 PERCENT \* FROM Customers;
* SELECT TOP 3 \* FROM Customers

WHERE Country='Germany';

* SELECT MIN(Price) AS SmallestPrice

FROM Products;

* SELECT MAX(Price) AS LargestPrice

FROM Products;

* SELECT COUNT(ProductID)

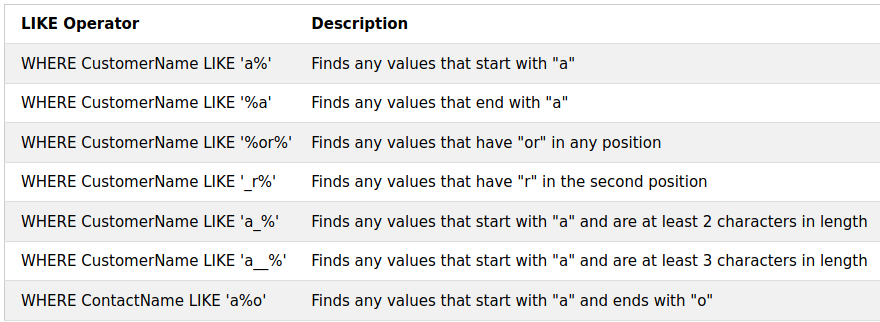
FROM Products;

* SELECT AVG(Price)

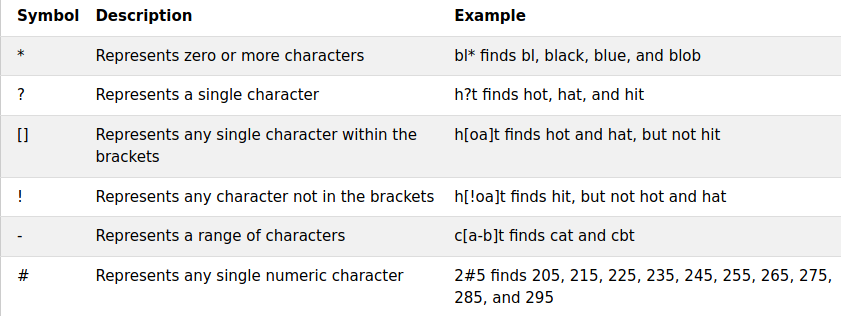
FROM Products;

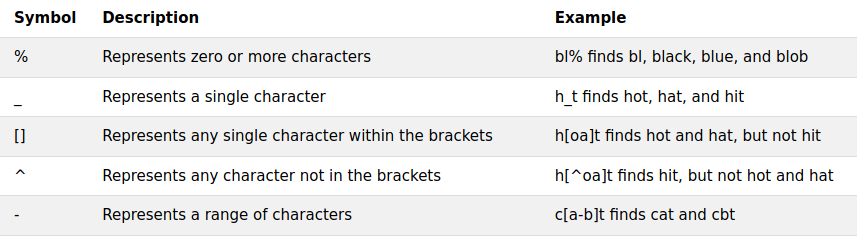
* SELECT SUM(Quantity)

FROM OrderDetails;

* 
* SELECT \* FROM Customers

WHERE CustomerName LIKE 'a%';

* 



* SELECT \* FROM Customers

WHERE City LIKE '[bsp]%';

bununla b,s veya p ile baslayan herhangi sehirler seciliyor.

* SELECT \* FROM Customers

WHERE City LIKE '[!bsp]%';

bununla da b,s veya p ile baslamayanlar listeleniyor!!!

* SELECT \* FROM Customers

WHERE Country IN ('Germany', 'France', 'UK');

* SELECT \* FROM Customers

WHERE Country IN (SELECT Country FROM Suppliers);

* SELECT \* FROM Products

WHERE Price BETWEEN 10 AND 20;

* SELECT \* FROM Products

WHERE Price BETWEEN 10 AND 20

AND CategoryID NOT IN (1,2,3);

* SELECT \* FROM Products

WHERE ProductName BETWEEN "Carnarvon Tigers" AND "Chef Anton's Cajun Seasoning"

ORDER BY ProductName;

* SELECT \* FROM Orders

WHERE OrderDate BETWEEN #07/01/1996# AND #07/31/1996#;

Bu iki tarih arasindaki tum degerleri alir!!!

* SELECT CustomerID AS ID, CustomerName AS Customer

FROM Customers;

* SELECT CustomerName AS Customer, ContactName AS [Contact Person]

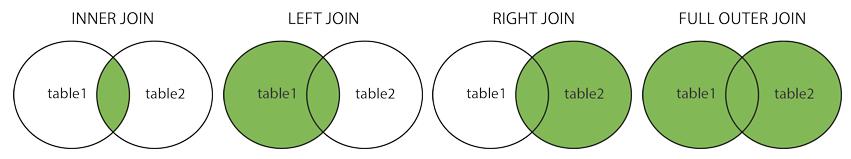
FROM Customers;

eger bosluk birakip yazmak istiyorsan bu sekilde kullanmalisin ya da ‘’ ile!!!

* SELECT CustomerName, Address + ', ' + PostalCode + ' ' + City + ', ' + Country AS Address

FROM Customers;

arti ile concat yapar ve ayni yere yan yana yazar!!!

* 
* SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate

FROM Orders

INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

* SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName

FROM ((Orders

INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)

INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);

* ##JOINLER ILE ILGILI NOTLAR##

Inner join ile atiyorum iki tabloyu birlestirecegiz ama sadece kesisimlerini aliyoruz aslinda. Join ile inner join ayni. Left join’i soyle dusun; iki tane kume var ve soldaki kumenin tum elemanlarini al ve sagdakiyle kesisimlerini yaz, ama soldaki her turlu olacak, kesisimleri yoksa null olarak gozukecek. Full join de aslinda left veya right join gibidir, hangi kumeyi once yazdiysan ona gore degisir.

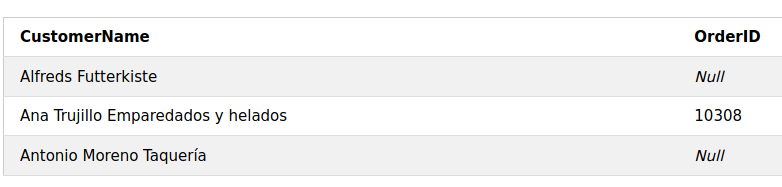
* SELECT Customers.CustomerName, Orders.OrderID

FROM Customers

FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID

ORDER BY Customers.CustomerName;

yukarida sari ile isaretledigim yer sol kume, digeri sag kume!!!



* SELECT City FROM Customers

UNION

SELECT City FROM Suppliers

ORDER BY City;

union customerlerdeki sehirlerle supplierlardaki sehirleri birlestirir, tekrar eden ifadeleri bir kere yazar sadece. eger tekrar tekrar yazmasini istiyorsan UNION ALL diye yazmalisin!!!

* SELECT COUNT(CustomerID), Country

FROM Customers

GROUP BY Country

HAVING COUNT(CustomerID) > 5;

* SELECT COUNT(CustomerID), Country

FROM Customers

GROUP BY Country

HAVING COUNT(CustomerID) > 5

ORDER BY COUNT(CustomerID) DESC;

* SELECT SupplierName

FROM Suppliers

WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price < 20);

* SELECT ProductName

FROM Products

WHERE ProductID = ANY

(SELECT ProductID

FROM OrderDetails

WHERE Quantity = 10);

* SELECT \* INTO CustomersBackup2017

FROM Customers;

* INSERT INTO Customers (CustomerName, City, Country)

SELECT SupplierName, City, Country FROM Suppliers;

* SELECT OrderID, Quantity,

CASE

WHEN Quantity > 30 THEN 'The quantity is greater than 30'

WHEN Quantity = 30 THEN 'The quantity is 30'

ELSE 'The quantity is under 30'

END AS QuantityText

FROM OrderDetails;

* SELECT ProductName, UnitPrice \* (UnitsInStock + ISNULL(UnitsOnOrder, 0))

FROM Products;

* CREATE PROCEDURE SelectAllCustomers

AS

SELECT \* FROM Customers

GO;

* EXEC SelectAllCustomers;
* CREATE PROCEDURE SelectAllCustomers @City nvarchar(30)

AS

SELECT \* FROM Customers WHERE City = @City

GO;

* EXEC SelectAllCustomers @City = 'London';
* CREATE PROCEDURE SelectAllCustomers @City nvarchar(30), @PostalCode nvarchar(10)

AS

SELECT \* FROM Customers WHERE City = @City AND PostalCode = @PostalCode

GO;

* EXEC SelectAllCustomers @City = 'London', @PostalCode = 'WA1 1DP';
* -- YAZINCA SONRA GELENLER COMMENT OLARAK ALGILANIYOR!!!

Multiline comments yazacaksan ‘/\*’ ve ‘\*/’ kullanmalisin!!!

* <> not equal to manasina gelir.
* CREATE DATABASE testDB;
* DROP DATABASE *databasename*;
* BACKUP DATABASE testDB

TO DISK = 'D:\backups\testDB.bak';

* BACKUP DATABASE *databasename*

TO DISK = ‘D:\backups\testDB.bak’

WITH DIFFERENTIAL;

sadece yapilan guncellemeleri alir!!!

* CREATE TABLE Persons (

PersonID int,

LastName varchar(255),

FirstName varchar(255),

Address varchar(255),

City varchar(255)

);

* CREATE TABLE TestTable AS

SELECT customername, contactname

FROM customers;

baska bir tablo kullanarak tablo olusturma!!!

* DROP TABLE Shippers;
* TRUNCATE TABLE *table\_name*;

tabloyu komple silmeden sadece icindeki datayi siler!!!

* ALTER TABLE Persons

ADD DateOfBirth date;

bu su anlama geliyor; ben persons tablosunda degisiklik yapmak istiyorum, dateofbirth diye bir sutun eklemek istiyorum ve bu sutun date turunde bir bilgi saklasin diyorum!!!

* CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255) NOT NULL,

Age int

);

not null diyere id nin null olamayacagini soyledik!!!

* CREATE TABLE Persons (

ID int NOT NULL UNIQUE,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int

);

unique ile id nin farkli degerler almasi gerektigini soyledik!!!

* CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int,

PRIMARY KEY (ID)

);

primary key null olamaz!!!

* CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int CHECK (Age>=18)

);

* CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int,

City varchar(255) DEFAULT 'Sandnes'

);

city ye default olarak sandnes degerini atiyor!!! (default)

* CREATE INDEX idx\_pname

ON Persons (LastName, FirstName);

lastname ve first name e idxpname indexini atadik!!!

* ALTER TABLE *table\_name*

DROP INDEX *index\_name*;

indexi silmek icin bunu yap!!!

* CREATE TABLE Persons (

Personid int IDENTITY(1,1) PRIMARY KEY,

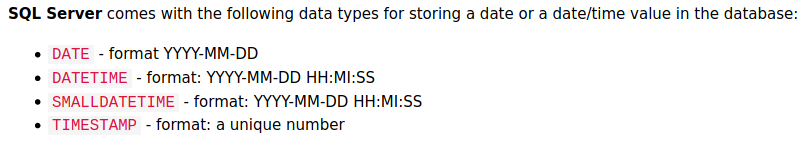
LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int

);

yukarida identity(1,1) diyerek bu degerin 1 ile baslayip otomatik bir sekilde birer birer artacagini belirttik. Yani tabloya yeni bir satir eklenince bu deger otomatik geliyor!!!



tarih ile saati hep ayri yazmaya calis, cunku veriyi cekmesi kolay olur boyle!!!

* CREATE VIEW [Brazil Customers] AS

SELECT CustomerName, ContactName

FROM Customers

WHERE Country = 'Brazil';

* SELECT \* FROM [Brazil Customers];

yukarida bir view olusturduk ve bu view i cagirdik!!!

* CREATE OR REPLACE VIEW [Brazil Customers] AS

SELECT CustomerName, ContactName, City

FROM Customers

WHERE Country = 'Brazil';

olusturulan view i guncelledik!!!

* DROP VIEW [Brazil Customers];

view i sildik!!!

<https://www.youtube.com/watch?v=rKwoBdlfo5g>