1. Show databases;
2. Create database Customers;
3. Use Customers;
4. Create Table

create table Customers(CustomerID int primary key , CustomerName varchar(50),ContactName varchar(60),Address varchar(50),City varchar(40),PostalCode int , Country varchar(50));

1. Show table;

Describe Customers;

1. Insert Data into tables

insert into Customers values(1,"Alfreds Futterkiste" , "Maria Anders","Obere Str. 57","Berlin",12209,"Germany");

insert into Customers values(2,"Ana Trujillo Emparedados y helados" , "Ana Trujillo","Avda. de la Constitución 2222","México D.F.",05021,"Mexico");

insert into Customers values(3,"Antonio Moreno Taquería" , "Antonio Moreno","Mataderos 2312","México D.F.",05023,"Mexico");

Query OK, 1 row affected (0.02 sec)

insert into Customers values(4,"Around the Horn" , "Thomas Hardy","120 Hanover Sq.","London",11111,"UK");

insert into Customers values(5,"Berglunds snabbköp" , "Christina Berglund","Berguvsvägen 8","Luleå",22222,"Sweden");

Query OK, 1 row affected (0.01 sec)

1. Show data of the Table

Select \* from Customers

1. Select command

SELECT CustomerName, City FROM Customers;

1. Distinct command

SELECT DISTINCT Country FROM Customers;

1. Where claue

SELECT \* FROM Customers  
WHERE Country='Mexico';

SELECT \* FROM Customers  
WHERE CustomerID=1;

1. And OR NOT operator

SELECT \* FROM Customers  
WHERE Country='Germany' AND City='Berlin';

SELECT \* FROM Customers  
WHERE City='Berlin' OR City='München';

SELECT \* FROM Customers  
WHERE NOT Country='Germany';

1. Order By

SELECT \* FROM Customers  
ORDER BY Country;

SELECT \* FROM Customers  
ORDER BY Country, CustomerName;

SELECT \* FROM Customers  
ORDER BY Country DESC;

1. Null values

SELECT CustomerName, ContactName, Address  
FROM Customers  
WHERE Address IS NULL;

14.Update commad

UPDATE Customers  
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'  
WHERE CustomerID = 1;

UPDATE Customers  
SET ContactName='Juan'  
WHERE Country='Mexico';

1. Delete Command

DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';

DELETE FROM Customers;

1. Aggregate function (min() , max() , count , avg() , sum());

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;

1. Like Operator

SELECT \* FROM Customers  
WHERE CustomerName LIKE 'a%';

SELECT \* FROM Customers  
WHERE CustomerName LIKE '%or%';

1. In operator

SELECT \* FROM Customers  
WHERE Country IN ('Germany', 'France', 'UK');

SELECT \* FROM Customers  
WHERE Country IN (SELECT Country FROM Suppliers);

1. Between Operator

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;

1. Aliases operatot

SELECT CustomerID AS ID, CustomerName AS Customer  
FROM Customers;

1. Join

Inner joins

SELECT column\_name(s)  
FROM table1  
INNER JOIN table2ON table1.column\_ name = table2.column\_name;

SELECT Orders.OrderID, Customers.CustomerName  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

Left join

SELECT column\_name(s)  
FROM table1  
LEFT JOIN table2ON table1.column\_name = table2.column\_name;

SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
ORDER BY Customers.CustomerName;

Right join

SELECT column\_name(s)  
FROM table1  
RIGHT JOIN table2ON table1.column\_name = table2.column\_name;

SELECT Orders.OrderID, Employees.LastName, Employees.FirstName  
FROM Orders  
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID  
ORDER BY Orders.OrderID;

Full outer join

SELECT column\_name(s)  
FROM table1  
FULL OUTER JOIN table2ON table1.column\_name = table2.column\_nameWHERE condition;

SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID  
ORDER BY Customers.CustomerName;

1. Union operator

SELECT column\_name(s) FROM table1  
UNION  
SELECT column\_name(s) FROM table2;

SELECT City FROM Customers  
UNION  
SELECT City FROM Suppliers  
ORDER BY City;

SELECT City, Country FROM Customers  
WHERE Country='Germany'  
UNION  
SELECT City, Country FROM Suppliers  
WHERE Country='Germany'  
ORDER BY City;

1. Group By operator

SELECT column\_name(s)  
FROM table\_name  
WHERE condition  
GROUP BY column\_name(s)ORDER BY column\_name(s);

SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country;

SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
ORDER BY COUNT(CustomerID) DESC;

1. Having clause;

SELECT column\_name(s)  
FROM table\_name  
WHERE condition  
GROUP BY column\_name(s)HAVING conditionORDER BY column\_name(s);

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;

1. Exist operator

SELECT column\_name(s)  
FROM table\_name  
WHERE EXISTS  
(SELECT column\_name FROM table\_name WHERE condition);

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

SELECT SupplierName  
FROM Suppliers  
WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price = 22);

1. Case Expression

CASE  
    WHEN condition1 THEN result1  
    WHEN condition2 THEN result2  
    WHEN conditionN THEN resultN  
    ELSE result  
END;

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;

1. Drop db

DROP DATABASE testDB;