1. SELECT \* FROM Customers / SELECT \* FROM table\_name;
2. SELECT CustomerName, City FROM Customers;
3. SELECT DISTINCT Country FROM Customers;
4. SELECT COUNT(DISTINCT Country) FROM Customers;
5. SELECT Count(\*) AS DistinctCountries  
   FROM (SELECT DISTINCT Country FROM Customers);

Where:

1. SELECT \* FROM Customers WHERE Country='Mexico';
2. SELECT \* FROM Customers WHERE CustomerID=1;

AND, OR, NOT

1. SELECT \* FROM Customers WHERE Country='Germany' AND City='Berlin';
2. SELECT \* FROM Customers WHERE City='Berlin' OR City='München';
3. SELECT \* FROM Customers WHERE Country='Germany' OR Country='Spain';
4. SELECT \* FROM Customers WHERE NOT Country='Germany';
5. SELECT \* FROM Customers WHERE Country='Germany' AND (City='Berlin' OR City='München');
6. SELECT \* FROM Customers WHERE NOT Country='Germany' AND NOT Country='USA';

ORDER BY

1. SELECT \* FROM Customers ORDER BY Country;
2. SELECT \* FROM Customers ORDER BY Country DESC;
3. SELECT \* FROM Customers ORDER BY Country, CustomerName;
4. SELECT \* FROM Customers ORDER BY Country ASC, CustomerName DESC;

INSERT INTO

1. INSERT INTO table\_name (column1, column2, column3, ...)  
   VALUES (value1, value2, value3, ...);
2. INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)  
   VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavang]\er', '4006', 'Norway');

NULL VALUES

1. SELECT column\_namesFROM table\_name WHERE column\_name IS NULL / (NOT NULL);
2. SELECT CustomerName, ContactName, Address FROM Customers WHERE Address IS NULL / (NO NULL);

UPDATE

1. UPDATE table\_name SET column1 = value1, column2 = value2, ... WHERE condition;
2. UPDATE Customers  
   SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'  
   WHERE CustomerID = 1;

DELETE

1. DELETE FROM table\_name WHERE condition;
2. DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';

SELECT TOP / LIMI

1. SELECT TOP number|*percent* column\_name(s) FROM table\_nameWHERE condition; [SQL]
2. SELECT column\_name(s) FROM table\_nameWHERE condition LIMIT number; [MY SQL]
3. SELECT TOP 3 \* FROM Customers; / SELECT \* FROM Customers LIMIT 3;
4. SELECT TOP 50 PERCENT \* FROM Customers; [SQL]

MIN / MAX / COUNT / AVG / SUM

1. SELECT MIN(Price) AS SmallestPrice FROM Products;
2. SELECT MAX(Price) AS LargestPrice FROM Products;
3. SELECT COUNT(ProductID) FROM Products;
4. SELECT AVG(Price) FROM Products;
5. SELECT SUM(Quantity) FROM OrderDetails;

LIKE

|  |  |
| --- | --- |
| **LIKE Operator** | **Description** |
| WHERE Name LIKE 'a%' | Finds any values that start with "a" |
| WHERE Name LIKE '%a' | Finds any values that end with "a" |
| WHERE Name LIKE '%or%' | Finds any values that have "or" in any position |
| WHERE Name LIKE '\_r%' | Finds any values that have "r" in the second position |
| WHERE Name LIKE 'a\_%' | Finds any values that start with "a" and are at least 2 characters in length |
| WHERE Name LIKE 'a\_\_%' | Finds any values that start with "a" and are at least 3 characters in length |
| WHERE Name LIKE 'a%o' | Finds any values that start with "a" and ends with "o" |
|  |  |

1. SELECT \* FROM Customers WHERE City LIKE '\_ondon';
2. SELECT \* FROM Customers WHERE City LIKE '[bsp]%'; //start with b, s, or p
3. SELECT \* FROM Customers WHERE City LIKE '[a-c]%'; //start with a to c
4. SELECT \* FROM Customers WHERE City LIKE '[!bsp]%'; //not start with b, s, or p

IN

1. SELECT column\_name(s) FROM table\_name WHERE column\_name IN (value1, value2, ...);
2. SELECT column\_name(s) FROM table\_name WHERE column\_name IN (*SELECT* STATEMENT);
3. SELECT \* FROM Customers WHERE Country IN ('Germany', 'France', 'UK');
4. SELECT \* FROM Customers WHERE Country NOT IN ('Germany', 'France', 'UK');
5. SELECT \* FROM Customers WHERE Country IN (SELECT Country FROM Suppliers);

BETWEEN

1. SELECT *column\_name(s)* FROM *table\_name* WHERE *column\_name* BETWEEN *value1* AND *value2;*
2. SELECT \* FROM Products WHERE Price BETWEEN 10 AND 20;
3. SELECT \* FROM Products WHERE Price NOT BETWEEN 10 AND 20;
4. SELECT \* FROM Products WHERE Price BETWEEN 10 AND 20 AND CategoryID NOT IN (1,2,3);

Aliases

1. SELECT column\_name(s) FROM table\_name AS alias\_name;
2. SELECT CustomerID AS ID, CustomerName AS Customer FROM Customers;
3. SELECT CustomerName, Address + ', ' + PostalCode + ' ' + City + ', ' + Country AS Address FROM Customers; [SQL]
4. SELECT CustomerName, CONCAT(Address,', ',PostalCode,', ',City,', ',Country) AS Address FROM Customers; [MY SQL]
5. SELECT o.OrderID, o.OrderDate, c.CustomerName  
   FROM Customers AS c, Orders AS o  
   WHERE c.CustomerName='Around the Horn' AND c.CustomerID=o.CustomerID;
6. SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerName  
   FROM Customers, Orders  
   WHERE Customers.CustomerName='Around the Horn' AND Customers.CustomerID=Orders.CustomerID;

JOIN

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

INNER JOIN

1. SELECT column\_name(s)  
   FROM table1  
   INNER JOIN table2ON table1.column\_name = table2.column\_name;
2. SELECT Orders.OrderID, Customers.CustomerName  
   FROM Orders  
   INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
3. 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);

LEFT JOIN

1. SELECT column\_name(s)  
   FROM table1  
   LEFT JOIN table2ON table1.column\_name = table2.column\_name;
2. SELECT Customers.CustomerName, Orders.OrderID  
   FROM Customers  
   LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
   ORDER BY Customers.CustomerName;

RIGHT JON

1. SELECT column\_name(s)  
   FROM table1  
   RIGHT JOIN table2ON table1.column\_name = table2.column\_name;
2. SELECT Orders.OrderID, Employees.LastName, Employees.FirstName  
   FROM Orders  
   RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID  
   ORDER BY Orders.OrderID;

FULL JOIN

1. SELECT column\_name(s)  
   FROM table1  
   FULL OUTER JOIN table2ON table1.column\_name = table2.column\_nameWHERE condition;
2. SELECT Customers.CustomerName, Orders.OrderID  
   FROM Customers  
   FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID  
   ORDER BY Customers.CustomerName;

UNION :

The UNION operator is used to combine the result-set of two or more SELECT statements. UNION select distinct value

1. SELECT City FROM Customers //union select distinct value  
   UNION  
   SELECT City FROM Suppliers  
   ORDER BY City;
2. SELECT City FROM Customers  
   UNION ALL //union all select all value  
   SELECT City FROM Suppliers  
   ORDER BY City;
3. SELECT City, Country FROM Customers  
   WHERE Country='Germany'  
   UNION ALL  
   SELECT City, Country FROM Suppliers  
   WHERE Country='Germany'  
   ORDER BY City;

GROUP BY

1. SELECT COUNT(CustomerID), Country  
   FROM Customers  
   GROUP BY Country;
2. SELECT Shippers.ShipperName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders  
   LEFT JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID  
   GROUP BY ShipperName;

HAVING

1. NB: Only include countries with more than 5 customers:  
   SELECT COUNT(CustomerID), Country  
   FROM Customers  
   GROUP BY Country  
   HAVING COUNT(CustomerID) > 5;  
   ORDER BY COUNT(CustomerID) DESC;

EXISTS

1. NB: Return lists the suppliers with a product price less than 20:  
   SELECT SupplierName  
   FROM Suppliers  
   WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price < 20);
2. TRUE and lists the suppliers with a product price equal to 22:  
   SELECT SupplierName  
   FROM Suppliers  
   WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price = 22);

ANY, ALL

1. SELECT ProductName  
   FROM Products  
   WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE Quantity = 10);

SQL DATABASE

1. CREATE DATABASE databasename;
2. DROP DATABASE databasename;
3. BACKUP DATABASE testDB TO DISK = 'D:\backups\testDB.bak';
4. CREATE TABLE Persons (  
       PersonID int,  
       LastName varchar(255),  
       FirstName varchar(255),  
       Address varchar(255),  
       City varchar(255)   
   );
5. DROP TABLE table\_name;
6. TRUNCATE TABLE table\_name;
7. ALTER TABLE Customers ADD Email varchar(255); //ADD NEW COLUMN
8. ALTER TABLE table\_name DROP COLUMN column\_name;
9. ALTER TABLE Customers DROP COLUMN Email;
10. CREATE TABLE Persons (  
        ID int NOT NULL UNIQUE,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255),  
        Age int  
    );
11. CREATE TABLE Persons (  
        ID int NOT NULL,  
        LastName varchar(255) NOT NULL,  
        FirstName varchar(255),  
        Age int,  
        PRIMARY KEY (ID)  
    );