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:

6. SELECT * FROM Customers WHERE Country='Mexico';
7. SELECT * FROM Customers WHERE CustomerID=1;

AND, OR, NOT

8. SELECT * FROM Customers WHERE Country='Germany' AND City='Berlin';
9. SELECT * FROM Customers WHERE City='Berlin' OR City='München';
10. SELECT * FROM Customers WHERE Country='Germany' OR Country='Spain';
11. SELECT * FROM Customers WHERE NOT Country='Germany';
12. SELECT * FROM Customers WHERE Country='Germany' AND (City='Berlin' OR City='München');
13. SELECT * FROM Customers WHERE NOT Country='Germany' AND NOT Country='USA';

ORDER BY

14. SELECT * FROM Customers ORDER BY Country;
15. SELECT * FROM Customers ORDER BY Country DESC;
16. SELECT * FROM Customers ORDER BY Country, CustomerName;
17. SELECT * FROM Customers ORDER BY Country ASC, CustomerName DESC;

INSERT INTO

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

NULL VALUES

20. SELECT *column_names* FROM *table_name* WHERE *column_name* IS NULL / (NOT NULL);
21. SELECT CustomerName, ContactName, Address FROM Customers WHERE Address IS NULL / (NO NULL);

UPDATE

22. UPDATE *table_name* SET *column1* = *value1*, *column2* = *value2*, ... WHERE *condition*;
23. UPDATE Customers
    SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
    WHERE CustomerID = 1;

DELETE

24. DELETE FROM *table_name* WHERE *condition*;
25. DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';

SELECT TOP / LIMI

26. SELECT TOP *number*|*percent column_name(s)* FROM *table_name* WHERE *condition*; [SQL]
27. SELECT *column_name(s)* FROM *table_name* WHERE *condition* LIMIT *number*; [MY SQL]
28. SELECT TOP 3 * FROM Customers; / SELECT * FROM Customers LIMIT 3;
29. SELECT TOP 50 PERCENT * FROM Customers; [SQL]

MIN / MAX / COUNT / AVG / SUM

30. SELECT MIN(Price) AS SmallestPrice FROM Products;
31. SELECT MAX(Price) AS LargestPrice FROM Products;
32. SELECT COUNT(ProductID) FROM Products;
33. SELECT AVG(Price) FROM Products;
34. 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"

35. SELECT * FROM Customers WHERE City LIKE '_ondon';
36. SELECT * FROM Customers WHERE City LIKE '[bsp]%'; //start with b, s, or p
37. SELECT * FROM Customers WHERE City LIKE '[a-c]%'; //start with a to c
38. SELECT * FROM Customers WHERE City LIKE '[!bsp]%'; //not start with b, s, or p

IN

39. SELECT *column_name(s)* FROM *table_name* WHERE *column_name* IN (*value1*, *value2*, ...);
40. SELECT *column_name(s)* FROM *table_name* WHERE *column_name* IN (*SELECT STATEMENT*);
41. SELECT * FROM Customers WHERE Country IN ('Germany', 'France', 'UK');
42. SELECT * FROM Customers WHERE Country NOT IN ('Germany', 'France', 'UK');
43. SELECT * FROM Customers WHERE Country IN (SELECT Country FROM Suppliers);

BETWEEN

44. SELECT *column_name(s)* FROM *table_name* WHERE *column_name* BETWEEN *value1* AND *value2;*
45. SELECT * FROM Products WHERE Price BETWEEN 10 AND 20;
46. SELECT * FROM Products WHERE Price NOT BETWEEN 10 AND 20;
47. SELECT * FROM Products WHERE Price BETWEEN 10 AND 20 AND CategoryID NOT IN (1,2,3);

Aliases

48. SELECT *column_name(s)* FROM *table_name* AS *alias_name;*
49. SELECT CustomerID AS ID, CustomerName AS Customer FROM Customers;
50. SELECT CustomerName, Address + ', ' + PostalCode + ' ' + City + ', ' + Country AS Address FROM Customers; [SQL]

51. SELECT CustomerName, CONCAT(Address,', ',PostalCode,', ',City,', ',Country) AS Address FROM Customers; [MY SQL]
52. 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;
53. SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerName
    FROM Customers, Orders
    WHERE Customers.CustomerName='Around the Horn' AND Customers.CustomerID=Orders.CustomerID;

JOIN

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

INNER JOIN

55. SELECT *column_name(s)*
    FROM *table1*
    INNER JOIN *table2*
    ON *table1.column_name* = *table2.column_name*;
56. SELECT Orders.OrderID, Customers.CustomerName
    FROM Orders
    INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
57. 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

58. SELECT *column_name(s)*
    FROM *table1*
    LEFT JOIN *table2*
    ON *table1.column_name* = *table2.column_name*;
59. SELECT Customers.CustomerName, Orders.OrderID
    FROM Customers
    LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    ORDER BY Customers.CustomerName;

RIGHT JON

60. SELECT *column_name(s)*
    FROM *table1*
    RIGHT JOIN *table2*
    ON *table1.column_name* = *table2.column_name*;
61. SELECT Orders.OrderID, Employees.LastName, Employees.FirstName
    FROM Orders
    RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
    ORDER BY Orders.OrderID;

FULL JOIN

62. SELECT *column_name(s)*
    FROM *table1*
    FULL OUTER JOIN *table2*
    ON *table1.column_name = table2.column_name*
    WHERE *condition*;
63. 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

64. SELECT City FROM Customers //union select distinct value
    UNION
    SELECT City FROM Suppliers
    ORDER BY City;
65. SELECT City FROM Customers
    UNION ALL //union all select all value
    SELECT City FROM Suppliers
    ORDER BY City;
66. SELECT City, Country FROM Customers
    WHERE Country='Germany'
    UNION ALL
    SELECT City, Country FROM Suppliers
    WHERE Country='Germany'
    ORDER BY City;

GROUP BY

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

HAVING

69. 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

70. 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);
```
71. 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

72.
```
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)
);
```