SQL CHEAT

1. SELECT * FROM Customers;

2. SELECT CustomerName, City FROM Customers

3. SELECT DISTINCT Country FROM Customers; **//// used to return only distinct (different) values.**

4. SELECT * FROM Customers
   WHERE Country='Mexico'; **//// used to extract only those records that fulfill a specified condition.**

5. SELECT * FROM Customers
   WHERE Country='Germany' AND City='Berlin'; //// **displays a record if all the conditions separated by AND are TRUE.**

6. SELECT * FROM Customers
   WHERE City='Berlin' OR City='München'; //// **displays a record if any of the conditions separated by OR is TRUE.**

7. SELECT * FROM Customers
   WHERE NOT Country='Germany' AND NOT Country='USA'; //// **displays a record if the condition(s) is NOT TRUE.**

8. SELECT * FROM Customers
   ORDER BY Country ASC, CustomerName DESC; //// **used to sort the result-set in ascending or descending order.**

9. INSERT INTO Customers (CustomerName, City, Country)
   VALUES ('Cardinal', 'Stavanger', 'Norway');

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

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

12. DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';

```
13.  SELECT MIN(Price) AS SmallestPrice
     FROM Products;

14. SELECT MAX(Price) AS LargestPrice
    FROM Products;

15. SELECT COUNT(ProductID)
    FROM Products;   ////  returns the number of rows that matches a specified criterion.

16. SELECT * FROM Customers
    WHERE Country IN ('Germany', 'France', 'UK'); ////  allows you to specify multiple values in a WHERE clause.

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

18. SELECT City FROM Customers
    UNION
    SELECT City FROM Suppliers
    ORDER BY City;  //// The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL.

19. SELECT COUNT(CustomerID), Country
    FROM Customers
    GROUP BY Country;   //// The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by
    one or more columns.

20. SELECT COUNT(CustomerID), Country
    FROM Customers
    GROUP BY Country
    ORDER BY COUNT(CustomerID) DESC;   //// The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group
    the result-set by one or more columns.

21.  SELECT * FROM Orders WHERE OrderDate='2008-11-11'
```