SQL

Below is a selection of demo table not the full table

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

1. select	`SELECT column1, column2, ...`
	`FROM table_name;`
	
	`SELECT * FROM table_name;`

2 select distinct	`SELECT DISTINCT Country FROM Customers;`
	
	`SELECT COUNT(DISTINCT Country) FROM Customers;`
	
	`SELECT Count(*) AS DistinctCountries`
	`FROM (SELECT DISTINCT Country FROM Customers);`

3 Where	`SELECT * FROM Customers` `WHERE Country='Mexico';`

4 And Or Not	`SELECT column1, column2, ...`
	`FROM table_name`
	`WHERE condition1 AND condition2 AND condition3 ...;`
	
	`SELECT column1, column2, ...`
	`FROM table_name`
	`WHERE condition1 OR condition2 OR condition3 ...;`
	
	`SELECT column1, column2, ...`
	`FROM table_name`
	`WHERE NOT condition;`
	
	`SELECT * FROM Customers`
	`WHERE Country='Germany' AND (City='Berlin' OR City='München');`

5. Order By	`SELECT column1, column2, ...`
	`FROM table_name`
	`ORDER BY column1, column2, ... ASC|DESC;`
	
	`SELECT * FROM Customers`
	`ORDER BY Country ASC, CustomerName DESC;`

6.insert into	`INSERT INTO table_name (column1, column2, column3, ...)`
	`VALUES (value1, value2, value3, ...);`
	
	`INSERT INTO table_name`
	`VALUES (value1, value2, value3, ...);`

	INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country) VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
7 Null	SELECT *column_names* FROM *table_name* WHERE *column_name* IS NULL; SELECT *column_names* FROM *table_name* WHERE *column_name* IS NOT NULL;
8 update	UPDATE *table_name* SET *column1* = *value1*, *column2* = *value2*, ... WHERE *condition*; UPDATE Customers SET ContactName = 'Alfred Schmidt', City= 'Frankfurt' WHERE CustomerID = 1;
9 delete	DELETE FROM *table_name* WHERE *condition*; DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
10select top	SELECT TOP *number*|*percent column_name(s)* FROM *table_name* WHERE *condition*; SELECT TOP 3 * FROM Customers; SELECT * FROM Customers LIMIT 3; SELECT TOP 50 PERCENT * FROM Customers; SELECT * FROM Customers WHERE Country='Germany' LIMIT 3;
11. Min max	SELECT MIN(*column_name*) FROM *table_name* WHERE *condition*; SELECT MAX(*column_name*) FROM *table_name* WHERE *condition*;
12 count, avg, sum	SELECT COUNT(*column_name*) FROM *table_name* WHERE *condition*; SELECT AVG(*column_name*) FROM *table_name* WHERE *condition*; SELECT SUM(*column_name*) FROM *table_name* WHERE *condition*;
13 LIKE	• % - The percent sign represents zero, one, or multiple characters • _ - The underscore represents a single character SELECT *column1, column2, ...* FROM *table_name* WHERE *columnN* LIKE *pattern*;

	LIKE Operator	**Description**
	WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
	WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
	WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
	WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
	WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 character
	WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 character
	WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"
	• `SELECT * FROM Customers WHERE CustomerName LIKE 'a%';`	

14 IN	`SELECT column_name(s) FROM table_name WHERE column_name IN (SELECT STATEMENT);`
	
	`SELECT * FROM Customers WHERE Country IN (SELECT Country FROM Suppliers);`	

15 between	`SELECT * FROM Products WHERE Price BETWEEN 10 AND 20;`	

16 Alias	`SELECT column_name AS alias_name FROM table_name;`
	
	`SELECT CustomerName, Address + ', ' + PostalCode + ' ' + City + ', ' + Country AS Address FROM Customers;`
	
	`SELECT CustomerName, CONCAT(Address,', ',PostalCode,', ',City,', ',Country) AS Address FROM Customers;`	

16 join	`SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate FROM Orders INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;`	
17 inner join	`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);`	
18 left join	`SELECT Customers.CustomerName, Orders.OrderID FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID ORDER BY Customers.CustomerName;`	
19 right join	`SELECT Orders.OrderID, Employees.LastName, Employees.FirstName FROM Orders RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID ORDER BY Orders.OrderID;`	
20 full join	`SELECT Customers.CustomerName, Orders.OrderID FROM Customers FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID ORDER BY Customers.CustomerName;`	
21 self join		

22 union	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;
23 group by	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 ORDER BY COUNT(CustomerID) DESC;
24 having	SELECT *column_name(s)* FROM *table_name* WHERE *condition* GROUP BY *column_name(s)* HAVING *condition* ORDER BY *column_name(s);* SELECT COUNT(CustomerID), Country FROM Customers GROUP BY Country HAVING COUNT(CustomerID) > 5 ORDER BY COUNT(CustomerID) DESC;
25 exist	SELECT *column_name(s)* FROM *table_name* WHERE EXISTS (SELECT *column_name* FROM *table_name* WHERE *condition*);
26 any all	The ANY and ALL operators are used with a WHERE or HAVING clause. The ANY operator returns true if any of the subquery values meet the condition. The ALL operator returns true if all of the subquery values meet the condition. SELECT *column_name(s)* FROM *table_name* WHERE *column_name operator* ANY (SELECT *column_name* FROM *table_name* WHERE *condition*); SELECT *column_name(s)* FROM *table_name* WHERE *column_name operator* ALL (SELECT *column_name* FROM *table_name* WHERE *condition*);
27 case	CASE WHEN *condition1* THEN *result1* WHEN *condition2* THEN *result2* WHEN *conditionN* THEN *resultN* ELSE *result* END;

```sql
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;
```

```sql
SELECT CustomerName, City, Country
FROM Customers
ORDER BY
(CASE
    WHEN City IS NULL THEN Country
    ELSE City
END);
```