SELF JOIN:

The SQL SELF JOIN is used to join a table to itself as if the table were two tables; temporarily renaming at least one table in the SQL statement.

Syntax

The basic syntax of SELF JOIN is as follows −

```
SELECT a.column_name, b.column_name...
FROM table1 a, table1 b
WHERE a.common_field = b.common_field;
```

Here, the WHERE clause could be any given expression based on your requirement

Example

Consider the following table.

CUSTOMERS Table is as follows.

```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

Now, let us join this table using SELF JOIN as follows −

```
SQL> SELECT  a.ID, b.NAME, a.SALARY
   FROM CUSTOMERS a, CUSTOMERS b
   WHERE a.SALARY < b.SALARY;
```

This would produce the following result −

```
+----+----------+---------+
| ID | NAME     | SALARY  |
+----+----------+---------+
|  2 | Ramesh   | 1500.00 |
|  2 | kaushik  | 1500.00 |
|  1 | Chaitali | 2000.00 |
|  2 | Chaitali | 1500.00 |
|  3 | Chaitali | 2000.00 |
|  6 | Chaitali | 4500.00 |
|  1 | Hardik   | 2000.00 |
|  2 | Hardik   | 1500.00 |
|  3 | Hardik   | 2000.00 |
|  4 | Hardik   | 6500.00 |
|  6 | Hardik   | 4500.00 |
|  1 | Komal    | 2000.00 |
|  2 | Komal    | 1500.00 |
|  3 | Komal    | 2000.00 |
|  1 | Muffy    | 2000.00 |
|  2 | Muffy    | 1500.00 |
|  3 | Muffy    | 2000.00 |
|  4 | Muffy    | 6500.00 |
|  5 | Muffy    | 8500.00 |
|  6 | Muffy    | 4500.00 |
+----+----------+---------+
```

1) Get all the result (student_id and name) from the table where student_id is equal, and course_id is not equal.

```
create table student(sid int , name varchar(20),course_id int);
 insert into student1 values(1,'adam',1),(2,'peter',2),(3,'brian',3);
insert into student1 values(1,'adam',2),(2,'shane',3);
select s1.sid,s1.name from student1 s1,student1 s2 where s1.sid=s2.sid and
s1.course_id<>s2.course_id;
```

```
mysql> select *from student;
+------+------+-------+
| sid  | cid  | since |
+------+------+-------+
|    1 | c1   |  2016 |
|    2 | c2   |  2017 |
|    3 | c2   |  2017 |
+------+------+-------+
3 rows in set (0.00 sec)

mysql> select *from student;
+------+------+-------+
| sid  | cid  | since |
+------+------+-------+
|    1 | c1   |  2016 |
|    2 | c2   |  2017 |
|    3 | c2   |  2017 |
+------+------+-------+
3 rows in set (0.00 sec)
```

```
mysql> select s1.sid,s1.name from student1 s1,student1 s2 where s1.sid=s2.sid and s1.course_id<>s2.course_id;
+------+-------+
| sid  | name  |
+------+-------+
|    1 | adam  |
|    2 | shane |
|    1 | adam  |
|    2 | peter |
+------+-------+
4 rows in set (0.00 sec)
```

2)

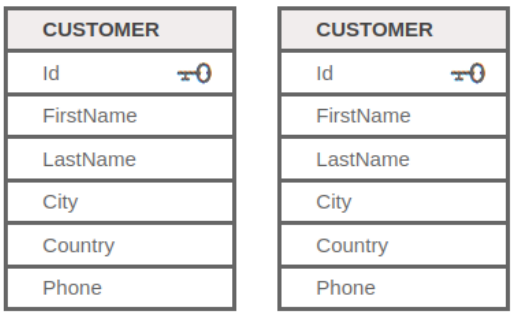
	employeeid	lastname	firstname	reportsto
▶	1	Davolio	Nancy	2
	2	Fuller	Andrew	NULL
	3	Leverling	Janet	2
	4	Peacock	Margaret	2
	5	Buchanan	Steven	2
	6	Suyama	Michael	5
	7	King	Robert	5
	8	Callahan	Laura	2
	9	Dodsworth	Anne	5

To display who reports to whom, you can join the employees table to itself as the following query:

```
SELECT
   concat(e.firstname, e.lastname) employee,
   concat(m.firstname, m.lastname) manager
FROM
   employees e
INNER JOIN
   employees m ON m.employeeid = e.reportsto;
```

	employee	manager
▶	NancyDavolio	AndrewFuller
	JanetLeverling	AndrewFuller
	MargaretPeacock	AndrewFuller
	StevenBuchanan	AndrewFuller
	MichaelSuyama	StevenBuchanan
	RobertKing	StevenBuchanan
	LauraCallahan	AndrewFuller
	AnneDodsworth	StevenBuchanan

3) Match customers that are from the same city and country




```
SELECT B.FirstName AS FirstName1, B.LastName AS LastName1,
       A.FirstName AS FirstName2, A.LastName AS LastName2,
       B.City, B.Country
  FROM Customer A JOIN Customer B
    ON A.Id <> B.Id
   AND A.City = B.City
   AND A.Country = B.Country
 ORDER BY A.Country
```

4) Which customers are located in the same state (column name is Region)? Type this statement in the SQL window:

```
SELECT DISTINCT c1.ContactName, c1.Address, c1.City, c1.Region

FROM Customers AS c1, Customers AS c2

WHERE c1.Region = c2.Region

AND c1.ContactName <> c2.ContactName

ORDER BY c1.Region, c1.ContactName;
```