SQL Full Outer Join

Last update on November 09 2019 06:55:13 (UTC/GMT +8 hours)

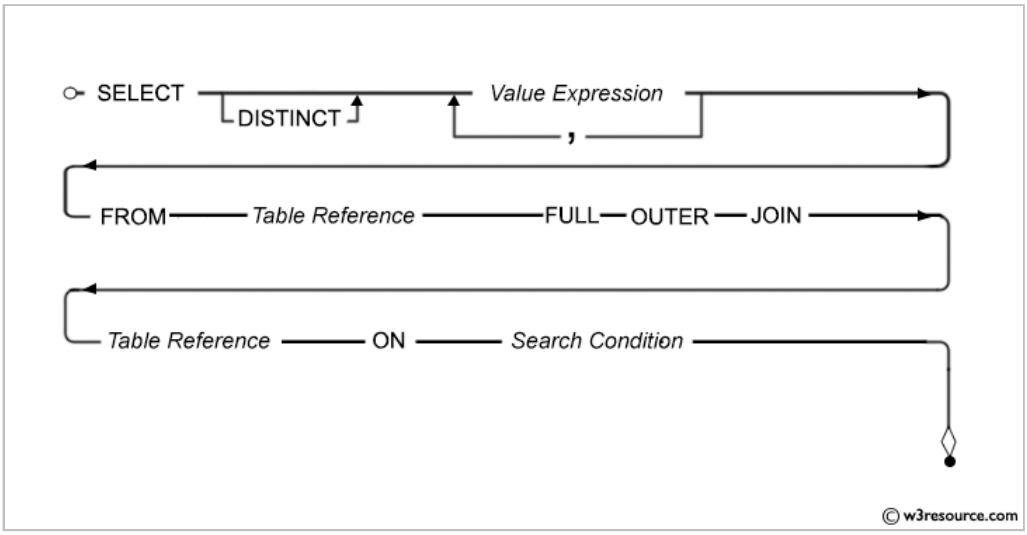
What is Full Outer Join in SQL?

In SQL the FULL OUTER JOIN combines the results of both left and right outer joins and returns all (matched or unmatched) rows from the tables on both sides of the join clause.

Syntax:

```
SELECT *
FROM table1
FULL OUTER JOIN table2
ON table1.column_name=table2.column_name;
```

Syntax diagram - FULL OUTER JOIN



Example: SQL FULL OUTER JOIN

Let's combine the same two tables using a full join.

table_A

A	M
1	m
2	n
4	o

table_B

A	N
2	p
3	q
5	r

SQL Code:

```
SELECT * FROM table_A
FULL OUTER JOIN table_B
ON table_A.A=table_B.A;
```

Output:

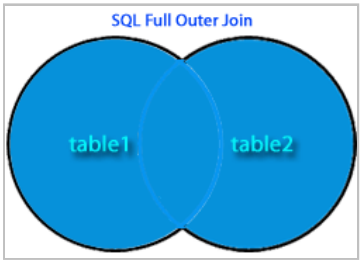
A	M	A	N
2	n	2	p
1	m	-	-
4	o	-	-
-	-	3	q
-	-	5	r

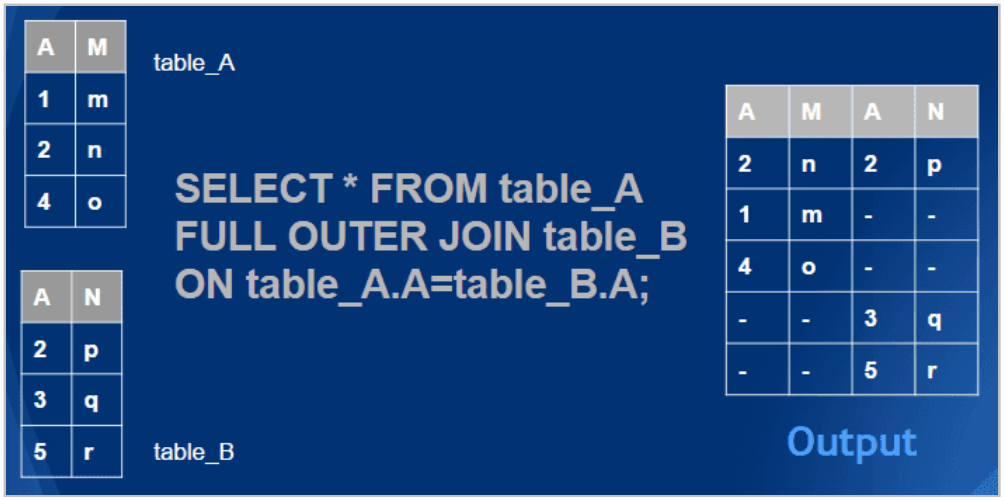
Because this is a full join, all rows (both matching and nonmatching) from both tables are included in the output. There is only one match between table table_A and table table_B, so **only one row of output displays values in all columns**. All remaining rows of output contain only values from table table_A or table table_B, with the remaining columns set to missing values

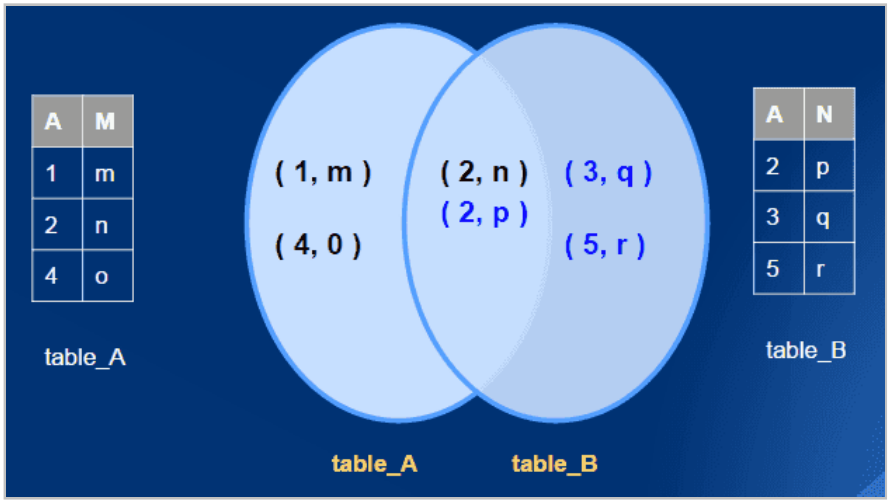
only one row of output displays values in all columns explain below -

A	M	A	N
2	n	2	p
1	m	-	-
4	o	-	-
-	-	3	q
-	-	5	r

Pictorial Presentation: SQL FULL OUTER JOIN







Example: SQL FULL OUTER JOIN between two tables

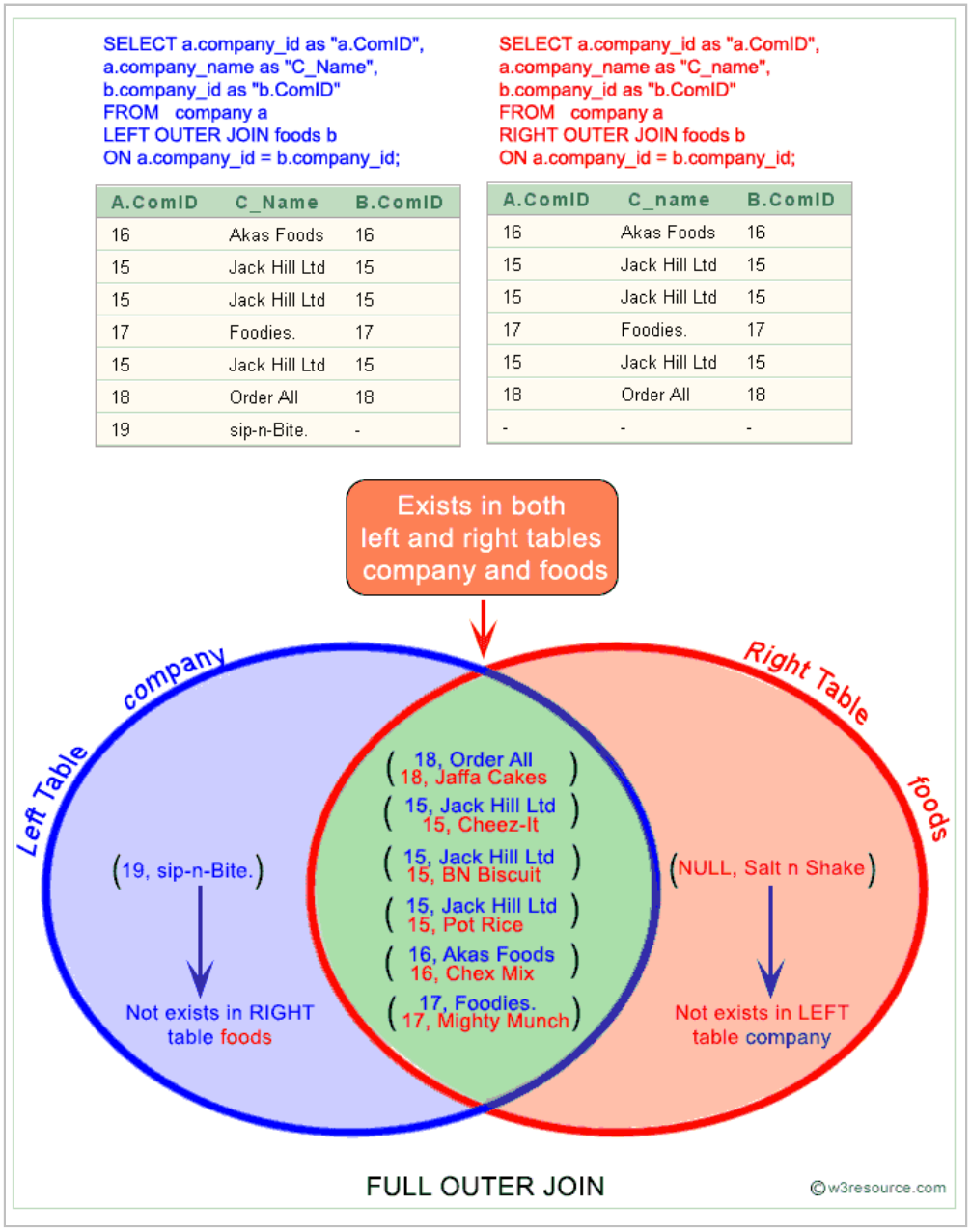
Here is an example of full outer join in SQL between two tables.

Sample table: foods

Sample table: company

As we know the FULL OUTER JOIN is the combination of the results of both LEFT OUTER JOIN and RIGHT OUTER JOIN, so, here we are going to describe how FULL OUTER JOIN perform internally.

Pictorial Presentation:



Here is the SQL statement which returns all rows from the 'foods' table and 'company' table using "FULL OUTER JOIN" clause.

SQL Code:

```
SELECT a.company_id AS "a.ComID",
a.company_name AS "C_Name",
b.company_id AS "b.ComID",
b.item_name AS "I_Name"
FROM   company a
FULL OUTER JOIN foods b
ON a.company_id = b.company_id;
```

Output:

```
a.ComID    C_Name                    b.ComID    I_Name
---------- ------------------------- ---------- -------------
16         Akas Foods                16         Chex Mix
15         Jack Hill Ltd             15         Cheez-It
15         Jack Hill Ltd             15         BN Biscuit
17         Foodies.                  17         Mighty Munch
15         Jack Hill Ltd             15         Pot Rice
18         Order All                 18         Jaffa Cakes
                                                Salt n Shake
19           sip-n-Bite.
```

FULL OUTER JOIN using WHERE clause

We can include a WHERE clause with a FULL OUTER JOIN to get return only those rows where no matching data between the joining tables are exist.

The following query returns only those company that have no matching food product in foods, as well as that food product in foods that are not matched to the listed company.

```
SELECT a.company_id AS "a.ComID",
a.company_name AS "C_Name",
b.company_id AS "b.ComID",
b.item_name AS "I_Name"
FROM   company a
FULL OUTER JOIN foods b
ON a.company_id = b.company_id
WHERE a.company_id IS NULL
OR b.company_id IS NULL
ORDER BY company_name;
```

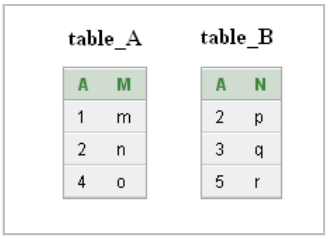
Output:

```
a.ComID    C_Name                    b.ComID    I_Name
---------- ------------------------- ---------- ---------------
19         sip-n-Bite.
                                                Salt n Shake
```

FULL OUTER JOIN using UNION clause

A UNION clause can be used as an alternate to get the same result as FULL OUTER JOIN

Here is the example:



Here is the SQL statement:

```
SELECT table_a.A,table_a.M,table_b.A,table_b.N
FROM table_A
FULL OUTER JOIN table_B
ON table_A.a=table_b.A
ORDER BY table_A.A;
```

FULL OUTER JOIN using LEFT and RIGHT OUTER JOIN and UNION clause

The following code is, the combination of LEFT OUTER JOIN and RIGHT OUTER JOIN and combined by, using UNION clause

```
SELECT table_a.A,table_a.M,table_b.A,table_b.N
FROM table_A
LEFT OUTER JOIN table_B
ON table_A.a=table_b.A
UNION
SELECT table_a.A,table_a.M,table_b.A,table_b.N
FROM table_A
RIGHT OUTER JOIN table_B
ON table_A.a=table_b.A;
```

Output:

A	M	A	N
1	m	-	-
2	n	2	p
4	o	-	-
-	-	3	q
-	-	5	r

Note: Outputs of the said SQL statement shown here is taken by using Oracle Database 10g Express Edition.