POSTGRESQL - UNIONS CLAUSE

The PostgreSQL **UNION** clause/operator is used to combine the results of two or more SELECT statements without returning any duplicate rows.

To use UNION, each SELECT must have the same number of columns selected, the same number of column expressions, the same data type, and have them in the same order but they do not have to be the same length.

Syntax:

The basic syntax of **UNION** is as follows:

```
SELECT column1 [, column2 ]
FROM table1 [, table2 ]
[WHERE condition]

UNION

SELECT column1 [, column2 ]
FROM table1 [, table2 ]
[WHERE condition]
```

Here given condition could be any given expression based on your requirement.

Example:

Consider following two tables, (a) COMPANY table is as follows:

```
testdb=# SELECT * from COMPANY;
 id | name  | age | address    | salary
----+-------+-----+------------+--------
  1 | Paul  |  32 | California |  20000
  2 | Allen |  25 | Texas      |  15000
  3 | Teddy |  23 | Norway     |  20000
  4 | Mark  |  25 | Rich-Mond  |  65000
  5 | David |  27 | Texas      |  85000
  6 | Kim   |  22 | South-Hall |  45000
  7 | James |  24 | Houston    |  10000
(7 rows)
```

(b) Another table is DEPARTMENT as follows:

```
testdb=# SELECT * from DEPARTMENT;
 id | dept        | emp_id
----+-------------+--------
  1 | IT Billing  |      1
  2 | Engineering |      2
  3 | Finance     |      7
  4 | Engineering |      3
  5 | Finance     |      4
  6 | Engineering |      5
  7 | Finance     |      6
(7 rows)
```

Now let us join these two tables using SELECT statement along with UNION clause as follows:

```
testdb=# SELECT EMP_ID, NAME, DEPT FROM COMPANY INNER JOIN DEPARTMENT
         ON COMPANY.ID = DEPARTMENT.EMP_ID
   UNION
     SELECT EMP_ID, NAME, DEPT FROM COMPANY LEFT OUTER JOIN DEPARTMENT
        ON COMPANY.ID = DEPARTMENT.EMP_ID;
```

This would produce the following result:

```
 emp_id | name  |  dept
--------+-------+-------------
      5 | David | Engineering
      6 | Kim   | Finance
      2 | Allen | Engineering
      3 | Teddy | Engineering
      4 | Mark  | Finance
      1 | Paul  | IT Billing
      7 | James | Finance
(7 rows)
```

The UNION ALL Clause:

The UNION ALL operator is used to combine the results of two SELECT statements including duplicate rows. The same rules that apply to UNION apply to the UNION ALL operator as well.

Syntax:

The basic syntax of **UNION ALL** is as follows:

```
SELECT column1 [, column2 ]
FROM table1 [, table2 ]
[WHERE condition]

UNION ALL

SELECT column1 [, column2 ]
FROM table1 [, table2 ]
[WHERE condition]
```

Here given condition could be any given expression based on your requirement.

Example:

Now, let us join above-mentioned two tables in our SELECT statement as follows:

```
testdb=# SELECT EMP_ID, NAME, DEPT FROM COMPANY INNER JOIN DEPARTMENT
        ON COMPANY.ID = DEPARTMENT.EMP_ID
   UNION ALL
     SELECT EMP_ID, NAME, DEPT FROM COMPANY LEFT OUTER JOIN DEPARTMENT
        ON COMPANY.ID = DEPARTMENT.EMP_ID;
```

This would produce the following result:

```
 emp_id | name  | dept
--------+-------+-------------
      1 | Paul  | IT Billing
      2 | Allen | Engineering
      7 | James | Finance
      3 | Teddy | Engineering
      4 | Mark  | Finance
      5 | David | Engineering
      6 | Kim   | Finance
      1 | Paul  | IT Billing
      2 | Allen | Engineering
      7 | James | Finance
      3 | Teddy | Engineering
      4 | Mark  | Finance
      5 | David | Engineering
      6 | Kim   | Finance
(14 rows)
```