PostgreSQL - NULL Values

The PostgreSQL **NULL** is the term used to represent a missing value. A NULL value in a table is a value in a field that appears to be blank.

A field with a NULL value is a field with no value. It is very important to understand that a NULL value is different from a zero value or a field that contains spaces.

Syntax

The basic syntax of using **NULL** while creating a table is as follows −

```
CREATE TABLE COMPANY(
   ID INT PRIMARY KEY     NOT NULL,
   NAME            TEXT    NOT NULL,
   AGE             INT     NOT NULL,
   ADDRESS         CHAR(50),
   SALARY          REAL
);
```

Here, **NOT NULL** signifies that column should always accept an explicit value of the given data type. There are two columns where we did not use NOT NULL. Hence, this means these columns could be NULL.

A field with a NULL value is one that has been left blank during record creation.

Example

The NULL value can cause problems when selecting data, because when comparing an unknown value to any other value, the result is always unknown and not included in the final results. Consider the following table, COMPANY having the following records −

```
ID          NAME        AGE         ADDRESS     SALARY
----------  ----------  ----------  ----------  ----------
1           Paul        32          California  20000.0
2           Allen       25          Texas       15000.0
3           Teddy       23          Norway      20000.0
4           Mark        25          Rich-Mond   65000.0
5           David       27          Texas       85000.0
6           Kim         22          South-Hall  45000.0
7           James       24          Houston     10000.0
```

Let us use the UPDATE statement to set few nullable values as NULL as follows −

```
testdb=# UPDATE COMPANY SET ADDRESS = NULL, SALARY = NULL where ID IN(6,7);
```

Now, COMPANY table should have the following records −

```
 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 |              |
  7 | James  |  24 |              |
(7 rows)
```

Next, let us see the usage of **IS NOT NULL** operator to list down all the records where SALARY is not NULL −

```
testdb=#  SELECT  ID, NAME, AGE, ADDRESS, SALARY
    FROM COMPANY
    WHERE SALARY IS NOT NULL;
```

The above given PostgreSQL statement will produce the following result −

```
 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
(5 rows)
```

The following is the usage of **IS NULL** operator which will list down all the records where SALARY is NULL −

```
testdb=#  SELECT  ID, NAME, AGE, ADDRESS, SALARY
        FROM COMPANY
        WHERE SALARY IS NULL;
```

The above given PostgreSQL statement will produce the following result −

```
 id | name   | age | address | salary
----+--------+-----+---------+--------
  6 | Kim    |  22 |         |
  7 | James  |  24 |         |
(2 rows)
```