Is Null – checks if a value is null or not.

PostgreSQL `Is null` clause statement syntax

Introduction to `NULL` and `IS NULL` operator

For example, if you have a `contacts` table that stores the first name, last name, email, and phone number of contacts. At the time of recording a contact, you may not know his or her phone number. To deal with this, you define the `phone` column as a nullable column and insert NULL into the `phone`column when you record the contact information.

```
CREATE TABLE contacts(
  id INT GENERATED BY DEFAULT AS IDENTITY,
  first_name VARCHAR(50) NOT NULL,
  last_name VARCHAR(50) NOT NULL,
  email VARCHAR(255) NOT NULL,
  phone VARCHAR(15),
  PRIMARY KEY (id)
);
```

The following statement inserts two contacts, one has a phone number and one does not:

```
INSERT INTO contacts(first_name, last_name, email, phone)
VALUES
   ('John','Doe','john.doe@example.com',NULL),
   ('Lily','Bush','lily.bush@example.com','(408-234-2764)');
```

To find the contact who does not have a phone number you may come up with the following statement:

```
SELECT id, first_name, last_name, email, phone FROM contacts WHERE phone = NULL
;

Or you can use it following query:

SELECT
id, first_name, last_name, email, phone FROM contacts WHERE phone IS NULL;
```

Here is the output:

id	first_name	last_name	email	phone
1	John	Doe	john.doe@example.com	(Null)

Is Null – checks if a value is null or not.

PostgreSQL `IS NOT NULL` operator

For example, to find the contact who does have a phone number, you use the following statement:

```
SELECT id, first_name, last_name, email, phone FROM contacts WHERE phone IS NOT NULL;
```

The output is:

id	first_name	last_name	email	phone
2	Lily	Bush	lily.bush@example.com	(408-234-2764)