College of Engineering, Trivandrum

Department of Computer Science and Engineering

CS333 APPLICATION SOFTWARE DEVELOPMENT LAB

LABORATORY REPORT 6

Data Constraints and Views

Student Name	Student ID
1. Justine Biju(S5)	170445(Roll No:37)

Submission Date : 29/07/2019

1 Introduction

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

1. NOT NULL
    (a) Ensures that a column cannot have a NULL value.
2. UNIQUE
    (a) Ensures that all values in a column are different.
3. PRIMARY KEY
    (a) A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table.
4. FOREIGN KEY
    (a) Uniquely identifies a row/record in another table
5. CHECK
    (a) Ensures that all values in a column satisfies a specific condition
6. DEFAULT
    (a) Sets a default value for a column when no value is specified
7. INDEX
    (a) Used to create and retrieve data from the database very quickly

In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.
You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

All these SQL queries are supported by PostgreSQL and can be implemented in a similar fashion.

2 Implementation in PostgreSQL

1. Lets create a table using some Data Constraints.
   'id' field can be PRIMARY KEY.
   'name' field can be NOT NULL.
   'ph_no' field can be UNIQUE.
   'child' field can have DEFAULT of 0.

   These can be done using the following query:

```
CREATE TABLE person (
id BIGSERIAL PRIMARY KEY,
name VARCHAR(50) NOT NULL,
ph_no VARCHAR(10) UNIQUE,
child INT DEFAULT 0);
```

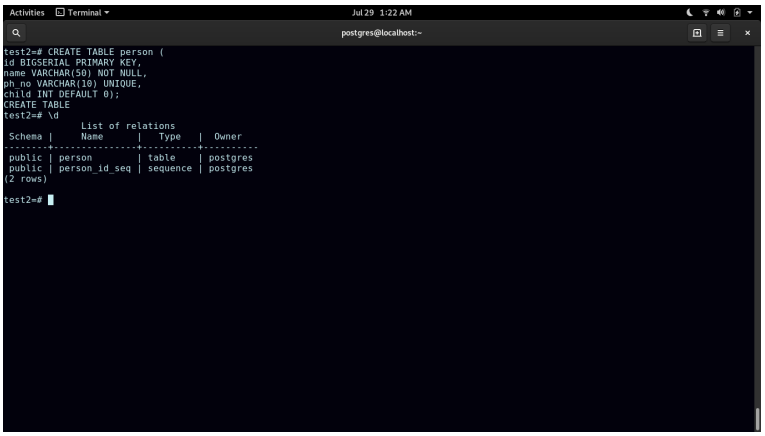

Figure 1: Using different constraints in SQL

2. Lets fill this table with values so that we can create views for this table.

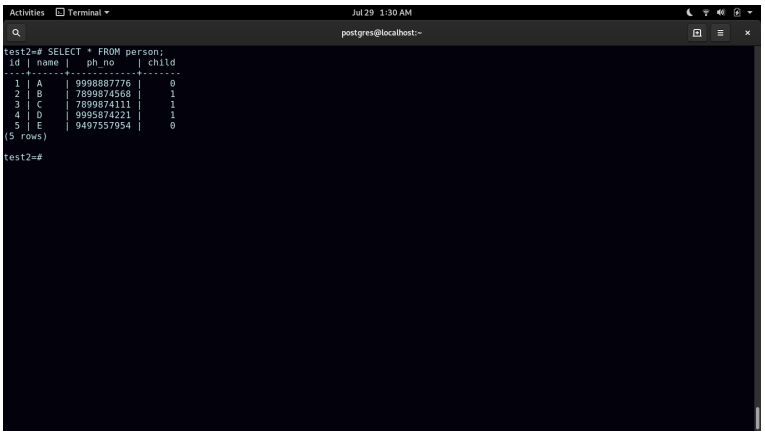

Figure 2: Records in table 'person'

3. Now let us create a view that contains only people that have no children.

```
CREATE VIEW "No Child" AS
SELECT * FROM person
WHERE child = 0;

SELECT * FROM "No Child";
```

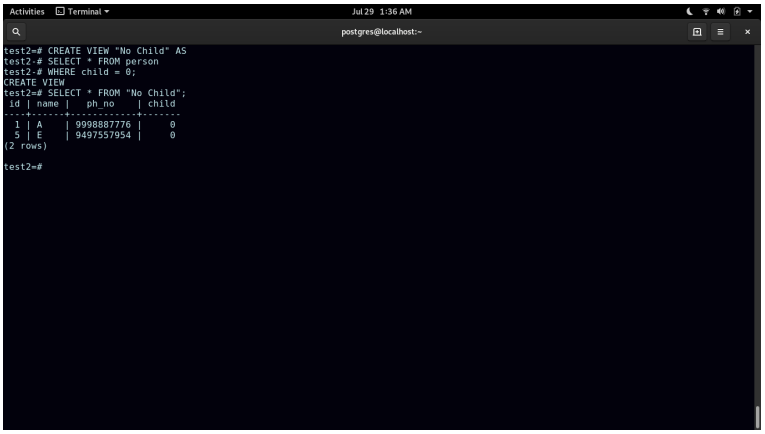

Figure 3: Creating a view "No Child"

4. We can drop a view using DROP query.

```
\d

DROP VIEW "No Child"

\d
```

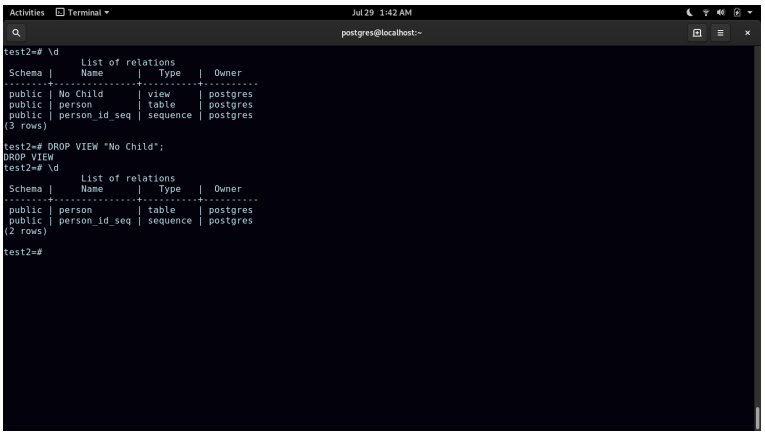

Figure 4: Dropping "No Child" view

3 Result

- Learned some important Data Constraints for PostgreSQL.
- Learned to create and manage views in PostgreSQL