

CHECK to Validate Data

by Sophia Tutorial



WHAT'S COVERED

This tutorial explores the use of the CHECK constraint to keep the values in a column limited to a set of criteria, in three parts:

1. Introducing The CHECK Constraint
2. Example Table
3. Error Messages

1. Introducing The CHECK Constraint

A CHECK is a unique constraint that verifies that the values being added to a column meet a specific requirement that we define. The CHECK constraint uses a Boolean expression to evaluate if the value is valid or not when it is being inserted into or updated in the column. If the check is valid, the database will then insert or update those values to the column. However, if the check is not valid, the database will reject the changes and raise an error in the database.

2. Example Table

The CHECK constraint is generally set up when we create the table. Let us look at an example of a CREATE TABLE statement with various CHECK constraints.

```
CREATE TABLE member (  
  member_id SERIAL PRIMARY KEY,  
  first_name VARCHAR (50),  
  last_name VARCHAR (50),  
  birth_date DATE CHECK (birth_date > '1900-01-01'),  
  joined_date DATE CHECK (joined_date > birth_date),  
  opt_in CHAR(1) CHECK (opt_in IN ('Y','N')),  
  membership_fee numeric CHECK(membership_fee > 0)  
);
```

Above, we have a table created with four CHECK constraints added with various data types. The first one is set based on the birth_date. It checks that the birth_date is after January 1, 1900. If we enter a birth date

before 01/01/1900, we will get an error message:

Query Results
Query failed because of: error: new row for relation "member" violates check constraint "member_birth_date_check"

The second CHECK constraint is based on the joined_date. It verifies that the member's joined_date is later than the birth_date. It wouldn't make sense to have a member joining prior to being born.

The third CHECK constraint is based on an opt_in column that is one character. It only allows a Y or N character. It could be argued that this could simply be set up as a Boolean. However, if the data is being sent in from a form, the value may need to be a specific character. We could also add other characters to be checked if needed.

The last CHECK constraint verifies the membership_fee is greater than zero.

3. Error Messages

You may have noticed in the error message that the constraint name was defined even though we did not set a name. In PostgreSQL, the constraint names are automatically created using the table name, column name, and type of constraint separated by underscores. If we wanted to have a specific name, we could replace the CHECK within the CREATE TABLE statement with a new line:

```
membership_fee numeric CONSTRAINT positive_fee CHECK(membership_fee > 0)
```

In this case, we name the CHECK constraint as positive_fee instead of the default member_membership_fee_check name.

We will explore how to add constraints after the table is created in an upcoming tutorial. In doing so, we can add more complex constraints over a series of columns.



TRY IT

Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then enter in one of the examples above and see how it works. Next, try your own choices for which columns you want the query to provide.

SUMMARY

The CHECK constraint is a good way to restrict values that a column can have at the database level.

Source: Authored by Vincent Tran