INSERT to Add Data

by Sophia

WHAT'S COVERED

This tutorial explores using the INSERT statement that adds data using an autoincremented primary key in two parts:

1. Introduction
2. Using the INSERT Statement to Add Data

1. Introduction

In a prior tutorial, we looked at creating tables using the SERIAL data type. The SERIAL data type is an auto-incrementing data type set to a column that is often used for the primary key. In PostgreSQL, this is done through a sequence that by default starts at 0 and increments by 1 each time the nextval function is called for the sequence. Within an INSERT statement, the column does not need to be included in the list, although it certainly could.

2. Using the INSERT Statement to Add Data

Let's explore a contact table:

```
CREATE TABLE contact( contact_id SERIAL, username VARCHAR(50), password VARCHAR(50) );
```

As a reminder, behind the scenes, a few steps are done to create the sequence for the contact_id:

```
CREATE SEQUENCE contact_contact_id_seq;

CREATE TABLE contact( contact_id integer NOT NULL DEFAULT nextval(contact_contact_id_seq), username VARCHAR(50), password VARCHAR(50) );

ALTER SEQUENCE contact_contact_id_seq
OWNED BY contact.contact_id;
```

Notice that the sequence table has the table name, an underscore, the column name, an underscore, and then seq as the sequence name. Once the table is created, this sequence is automatically created.

Using the method we explored in the prior tutorial, we can insert into this table using the following:

```
INSERT INTO contact (contact_id,username,password)
VALUES (nextval('contact_contact_id_seq'),'sophia','Password1');
```

Notice that the value for the contact_id uses the function nextval and passes in the parameter of the sequence.

If we query the table, we should see:

Query Results

Row count: 1

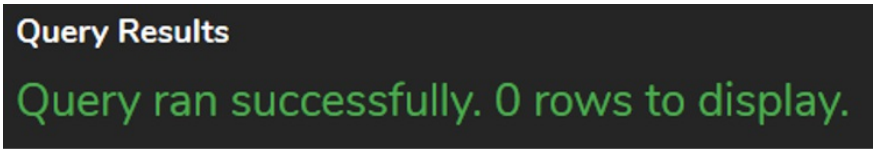
contact_id	username	password
1	sophia	Password1

However, it can be quite cumbersome to always write out the entire function to get the next value. Simply by inserting a row into the table without specifying the contact_id, the nextval function is automatically called. For example, if we ran the following insert statement:

```
INSERT INTO contact (username, password)
VALUES ('Mustang','Password3');
```

```
INSERT INTO contact (username,password)
VALUES ('roda','Password99');
```

You should see that the contact_id automatically increments:



Specific to PostgreSQL, you can also use the keyword DEFAULT for the value if you wanted to pass in the contact_id as well:

```
INSERT INTO contact (contact_id,username,password)
VALUES (DEFAULT,'Caloric','Password77');
```

Query Results

Row count: 4

contact_id	username	password
1	sophia	Password1
2	Mustang	Password3
3	roda	Password99
4	Caloric	Password77

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

We can add rows into a table using the INSERT statement with an auto-incremented column without passing in any values.

Source: Authored by Vincent Tran