ALTER TABLE to Change Columns: Data Characteristics

by Sophia

WHAT'S COVERED

In this lesson, you will use the ALTER TABLE command to change a column's size. This lesson will be explored in two parts. Specifically, this lesson will cover:

1. Why Column Size Might Need to Change
2. Changing Column Size

1. Why Column Size Might Need to Change

Normally a column's size is defined when you create the table. However, there are cases where a column's size might need to change after the table is already in use. For example, you might make a column larger to accommodate larger entries because you underestimated the size of the entries to be made in that column. Conversely, you might realize that you have overestimated a column's size need and want to make the database more efficient by trimming column sizes to match the data you are actually storing in them.

HINT

Decreasing a column width is allowed, but only if doing so would not result in data loss. For example, if you were going from a size of 50 to a size of 25, and all entries in that column were 25 characters or fewer, it would work fine. But if one entry were 26 characters, an error would appear. To fix this, you would need to change the 26-character value to 25 or fewer characters.

2. Changing Column Size

There is no separate SQL command specifically for changing the size of a column; instead, we use the same TYPE operator you learned about in the previous lesson.

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

```
ALTER TABLE <tablename>
ALTER COLUMN <columnname>
TYPE <newdatatype>;
```

We can keep the same data type but just change the column size or use a smaller/larger version of the data type if one exists.

For example, let's take a look at a basic registration table used to capture individuals registering for an event:

```
CREATE TABLE registration(
registration_id int PRIMARY KEY,
first_name VARCHAR(10),
last_name VARCHAR(10),
email VARCHAR(30),
fee NUMERIC(4,2)
);
```

Let's add some sample data. We will cover this in a later lesson:

```
NSERT INTO registration VALUES (1,'Michelle','Pippen','mpippen@a.com',9.99);
INSERT INTO registration VALUES (2, 'Santana','Smith','smith@b.com',9.99);
```

No error should occur, and if we query the table, we should see the two rows:

Query Results

Row count: 2

registration_id	first_name	last_name	email	fee
1	Michelle	Pippen	mpippen@a.com	9.99
2	Santana	Smith	smith@b.com	9.99

However, suppose that a new individual is trying to register with a last_name of more than 10 characters. Consider if we tried to insert a record with a value longer than 10 for the last name, like this:

```
INSERT INTO registration VALUES (3, 'Joseph','Rudy-Potter','jpotter@c.com',9.99);
```

We should get an error like this:

Query Results

Query failed because of: error: value too long for type character varying(10)

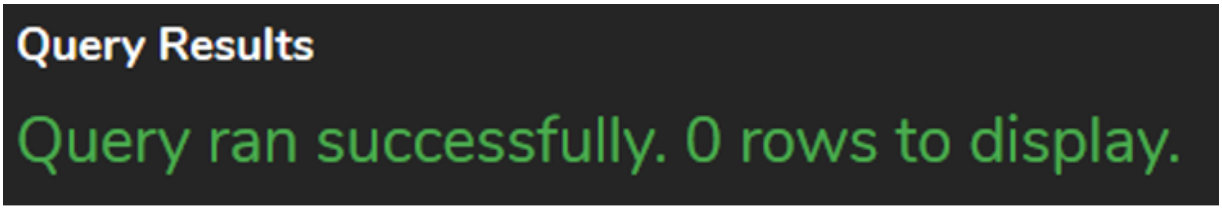
We could simply count the number of characters and set the length of the variable to that value, but it would be better to anticipate future potential sizes to avoid having to make this adjustment again later. It is better to err

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

on the higher side rather than the lower side. We can go ahead and change the length of the last name to 50 characters:

```
ALTER TABLE registration
ALTER COLUMN last_name
TYPE VARCHAR(50);
```

Now, if we run the insert statement again, it should run successfully:



We can make changes to more than one column at a time. For example, the first name and email should probably also have a larger size. We can combine them together by listing each ALTER COLUMN statement separated by a comma:

```
ALTER TABLE registration
ALTER COLUMN first_name
TYPE VARCHAR(50),
ALTER COLUMN email
TYPE VARCHAR(100);
```

In looking at the fee, we are limited to four digits, with two digits after the decimal point. This means the maximum value that could be inserted is 99.99. To change this, we can use the same statement to increase the precision of the fee to six digits:

```
ALTER TABLE registration
ALTER COLUMN fee
TYPE numeric(6,2);
```

This will allow values up to 9999.99 to be inserted into the table.

It is important to note that we cannot alter a column's data type that has a foreign key reference to another table. If there is a foreign key reference, the data type and size have to be the same as the primary key to which it is related. For example, if we tried to alter the artist_id in the album table, due to the foreign key to the artist_id in the artist table, we would get the following error:

```
ALTER TABLE album
ALTER COLUMN artist_id
TYPE VARCHAR (100);
```

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

Query Results

Query failed because of: error: foreign key constraint "album_artist_id_fkey" cannot be implemented

WATCH

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

In this lesson, you learned **why it may be useful to change a column's size** in a table that is already in use. You then learned how to use the ALTER TABLE statement in PostgreSQL to increase or decrease a column's size.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND Faithe Wempen (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR TERMS OF USE.

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.