

# ALTER TABLE to Change Columns: Data Type

by Sophia Tutorial



## WHAT'S COVERED

This tutorial explores using the ALTER TABLE statement to change the data type of a column in a table:

1. Changing Columns
2. A Second Example

## 1. Changing Columns

Although it is rare to do so once data has been inserted into a column, there are times when we may want to change the data type of a column. When we make a change to the data type, PostgreSQL will cast the values to the new data type. If casting the data type to the new one fails, the database will issue an error.

The structure of the command looks like the following:

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

Let's start with a basic contact table that was created to handle requests for an application:

```
CREATE TABLE contact( contact_id int PRIMARY KEY, username VARCHAR(50), password VARCHAR(50), opt_in int );
```

Let's add some sample data to the table. We will cover these statements in a future tutorial:

```
INSERT INTO contact VALUES(1,'bob798','mypass1',1);  
INSERT INTO contact VALUES(2,'jen558','mypass2',1);  
INSERT INTO contact VALUES(3,'rand058','mypass3',1);
```

In this scenario, the organization originally wanted to have the application pass in 0 to opt-out and 1 to opt-in.

## 2. A Second Example

However, perhaps the organization decides this is too challenging to decipher and instead wants to convert the data type in the opt\_in column to a character. To do so, we would execute the following command:

```
ALTER TABLE contact
ALTER COLUMN opt_in
TYPE CHAR(1);
```

Changing from an integer to a character here does not create an error. Let's add more sample data into the table using the updated data type. Notice that we are now using Y instead of 1 in the opt\_in column:

```
INSERT INTO contact VALUES(4,'jeff768','mypass4','Y');
INSERT INTO contact VALUES(5,'sandra547','mypass5','Y');
INSERT INTO contact VALUES(6,'roberto9128','mypass6','Y');
```

However, if we tried to convert opt\_in back to an int, we should get an error:

```
ALTER TABLE contact
ALTER COLUMN opt_in
TYPE int;
```

```
Query Results
Query failed because of: error: column "opt_in" cannot be cast automatically to type integer
```

This is because we have the character Y in the opt\_in table in the last three records:

```
Query Results
Row count: 6
```

contact_id	username	password	opt_in
1	bob798	mypass1	1
2	jen558	mypass2	1
3	rand058	mypass3	1
4	jeff768	mypass4	Y
5	sandra547	mypass5	Y
6	roberto9128	mypass6	Y

This is an issue to consider when you change the data type. Even though changing from an integer to a character did not create an error, if the opt\_in had a value of two digits, we'd also run into an issue as the type that it was converted to was set to one character. When we change the data type, it is important to consider both the data that the table already contains, and what it should be changed to. If there isn't any data at all in the table yet, it won't be an issue to make those data conversions using the ALTER TABLE command.

## Video Transcription

With the ALTER TABLE statement you can actually modify the data type associated with a particular column for a table. It is important to note that once you actually have data within table, it can become a little bit more complex, in this case here. You have to be careful with the modifications.

So here we're taking a look at the track table. We're looking specifically at the track ID, name, album ID, and bytes. If we look at the data for the name column itself, we'll see that it's all text-based. If we try to modify this to a different data type like an integer for example, we should be presented with an error. If there was no data within the table, you can certainly make the modification and it wouldn't have an issue, but because there was data in there it couldn't actually cast that data to an integer type.

Let's go ahead and try to modify the album ID in the same way too. It's currently set up as integer if we take a look at the table itself. And if we try to modify it to a numeric value, we also are going to get an error, in this case here, because there is actually a foreign key associated with the album ID from the album table. However, if we take a look at the bytes, it's currently set up as integer as well. But in this case here, we can make a modification to it because there's no reference to it and all the data types can be modified to of our chart without any issues.



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

Using the ALTER TABLE command, you can change the data type of columns; however, it does depend on what data is already inserted into the column.

Source: Authored by Vincent Tran