DELETE FROM to Remove Row

by Sophia Tutorial

WHAT'S COVERED

This tutorial explores using the DELETE statement to remove rows in a table in two parts:

1. Using the DELETE FROM Statement
2. Foreign Keys
3. Deleting Multiple Rows

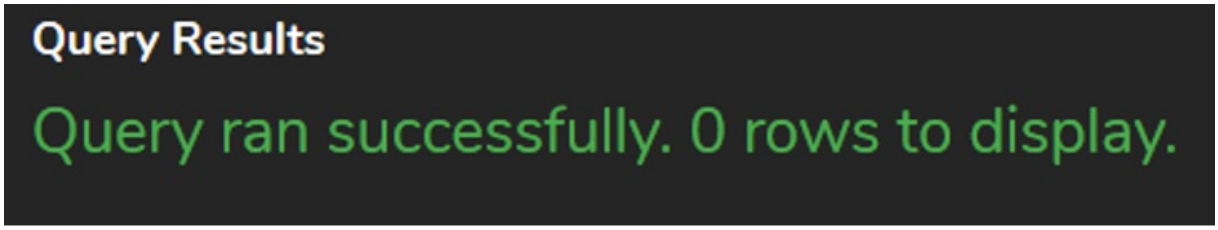
1. Using the DELETE FROM Statement

The DELETE statement is a relatively easy statement to learn, as it uses the same WHERE clause that you have seen with the SELECT and UPDATE statements. The syntax for the DELETE statement looks like the following:

DELETE FROM <tablename>
WHERE <condition>;
In the syntax, we first have the name of the table that we want to remove rows from listed after the DELETE FROM keywords. Note that we can only delete from one table at a time. Then we use the WHERE clause to specify which rows from the table we want to delete. Like the UPDATE statement, if we do not use the WHERE clause, the DELETE statement will end up deleting all the rows in our table.

DELETE FROM invoice_line;

SELECT *
FROM invoice_line;



Oops! Although there are times when we may want this, this is usually not what we want to do with our DELETE statements.

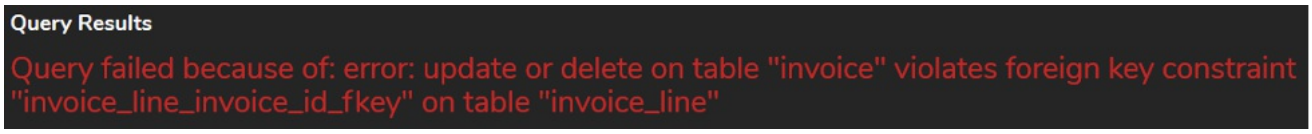
In PostgreSQL, we can also use the RETURNING keyword to return the rows that have been removed from

the table. However, it's a better practice to test your WHERE clause with a SELECT statement before running the DELETE statement.

2. Foreign Keys

It's important to note that we cannot delete rows that are referenced by the foreign key from other tables. For example, if we tried to delete the invoice where the invoice_id is equal to 1:

```
DELETE FROM invoice
WHERE invoice_id = 1;
```

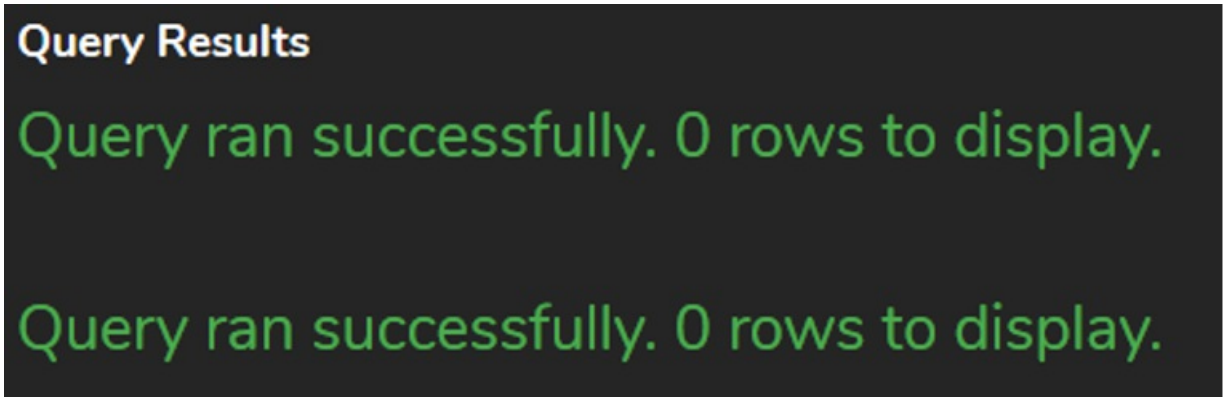


We are not able to delete it, as there is a foreign key from the invoice_line table. This is similar to what we saw with the DROP table, and we must delete data in the same order.

```
DELETE FROM invoice_line
WHERE invoice_id = 1;
```

```
DELETE FROM invoice
WHERE invoice_id = 1;
```

This time, we were successful in deleting the invoice_id = 1 from both tables. This is important to avoid having rows of data that have linking issues.



3. Deleting Multiple Rows

We can also use the WHERE clause to delete multiple rows and options at the same time. For example, we could remove all the invoices between 1-10:

```
DELETE FROM invoice_line
WHERE invoice_id BETWEEN 1 AND 10;
```

```
DELETE FROM invoice
WHERE invoice_id BETWEEN 1 AND 10;
```

We could also base it on a specific range:

```
DELETE FROM invoice_line
WHERE invoice_id > 50;

DELETE FROM invoice
WHERE invoice_id > 50;
```

Video Transcription

[MUSIC PLAYING] The DELETE statement can be utilized to remove rows from tables. Let's take a look at the contact table that we initially created. And we want to go ahead and delete any contact ID that's less than 10. So the framework of the statement is very simple. It looks like this. It looks like DELETE FROM as a keywords. Then we include the table name.

And then we have the WHERE clause. You'll see that WHERE clause repeated from the SELECT statement and from the INSERT, as well as UPDATE statements where the contact ID, which is the column is going to be less than 10. Once we do that, I will select from the table. You'll notice in this case here, that the real count is now only set at four, rather than 13 that we had previously, meaning that has been removed completely.

One important thing to note is that if you have the DELETE statement without the WHERE clause, it'll end up deleting all the rows from the table, which is typically what we don't want. So you do want to be careful in this case here. So for example, if I went ahead and ran that and tried to select, no rows would be included in the tables.

Another key piece to make note as well, is that the deletion of rows does depend on foreign keys. So if we go ahead and take a look at the album table in this case, we'll have some references to foreign keys as well. So if we tried to delete the album ID equals 1, we should have an error in this case here, being that we have a couple of other tables that references to that album ID.

[MUSIC PLAYING]

So you'll notice in this case here, on the track table, it has a foreign key on the album ID within the track table to the album table's album ID. So because of that, we actually have error in which we have to delete the items from track table first before we can delete from the album table. But then we also have to make sure that anything that's referencing that particular track is also removed. And it needs to go in the same order that we've seen previously when we went through the drop table process.

[MUSIC PLAYING]

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.

NOTE: It is okay to practice deleting rows in the learning SQL tool. If you need to recover missing data or tables from the original database, just reload the browser page for the tool or come back to this tutorial and launch the database again.

SUMMARY

The DELETE statement provides a means to remove rows from tables.

Source: Authored by Vincent Tran