

# DELETE FROM to Remove Row

by Sophia



## WHAT'S COVERED

In this lesson, you will use the DELETE statement to remove rows in a table, in three parts. Specifically, this lesson will cover:

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

## 1. Using the DELETE FROM Statement

The DELETE FROM statement is relatively easy to learn, as it uses the same WHERE clause you have seen with the SELECT and UPDATE statements. The syntax for the DELETE FROM 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.



### KEY CONCEPT

Like the UPDATE statement, if we do not use the WHERE clause, the DELETE FROM statement will end up deleting all the rows in our table.

```
DELETE FROM invoice_line;  
SELECT *  
FROM invoice_line;
```

## Query Results

Query ran successfully. 0 rows to display.

This is a very real issue when working with the delete command. Much like DROP TABLE, you want to ensure you have a backup of all the data before working with the DELETE FROM statement. Although there are times when we may want this, this is usually different from what we want to do with our DELETE FROM 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 FROM 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, we would get the following error:

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

### Query Results

Query failed because of: error: update or delete on table "invoice" violates foreign key constraint "invoice\_line\_invoice\_id\_fkey" on table "invoice\_line"

We cannot delete it, as there is a foreign key from the invoice\_line table. This is similar to what we saw with the DROP TABLE; we must delete data in the same order as shown below.

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

## Query Results

Query ran successfully. 0 rows to display.

Query ran successfully. 0 rows to display.

By deleting first from the foreign key side, and then from the primary key side, we successfully deleted the `invoice_id = 1` from both tables. This is important in order to avoid having rows of data that have linking issues.

### 3. Deleting Multiple Rows

We can use the `WHERE` clause to specify multiple rows for deletion. 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;
```



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 how to delete specific rows from tables with the **DELETE FROM statement** based on WHERE conditions. It's important to include a WHERE clause because if you don't, you will delete every row in the table. You also learned that you cannot delete rows that are referenced by the **foreign key** from other tables; you must delete the corresponding rows in the foreign key table first. Finally, you learned that you can use operators such as BETWEEN or math operators such as = and > to specify **multiple rows for deletion**.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND FAITH WEMPEN (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR [TERMS OF USE](#).