

Foreign Keys & Altering Tables

by Sophia



WHAT'S COVERED

This tutorial explores using the ALTER TABLE command to add a foreign key in two parts:

1. ALTER TABLE
2. ON DELETE clause

1. ALTER TABLE

An ALTER TABLE statement can be used to add a foreign key to a table. Although typically the foreign key is added during a CREATE TABLE statement, there are instances when it can be beneficial to add foreign keys after the table has been created and the data has been loaded.

When we use a CREATE TABLE statement, we do not have to define the foreign key constraint name, as PostgreSQL and other databases will assign an auto-generated name. The basic syntax to add a foreign key constraint to an existing table looks like this:

```
ALTER TABLE <childtablename>
ADD CONSTRAINT <constraintname>
FOREIGN KEY (<fk_column>)
REFERENCES <parenttablename> (<parent_key_column>);
```

Given the following tables:

```
CREATE TABLE representative ( representative_id INT PRIMARY KEY, first_name VARCHAR (30) NOT NULL, last_name VARCHAR (30) NOT NULL );
```

```
CREATE TABLE department ( department_id INT PRIMARY KEY, department_name VARCHAR (100) NOT NULL, manager_id INT );
```

We can add a foreign key to the department table for the manager_id by doing:

```
ALTER TABLE department
ADD CONSTRAINT fk_manager
FOREIGN KEY (manager_id)
REFERENCES representative (representative_id);
```

It is not that different from the syntax used in the CREATE TABLE statement, but the ordering of the statements is important.

2. ON DELETE clause

There are additional clauses that can be set with the ALTER TABLE statement. We can add ON DELETE SET NULL, which will set the value to null in the child table if the parent record is deleted. This means that the foreign key column would have to allow NULL values to be used.

We can also use the ON DELETE SET DEFAULT clause and pass in a specific value. This means that if the parent record is deleted, the child record would update to a set value. For example, if a manager is deleted, the department could automatically be moved under a specific generic employee and reallocated later on.

The ON DELETE CASCADE clause can be a dangerous one, but it has common uses. Using this clause, a delete on a parent record would also delete all of the referencing rows in the child table. For example, in our database, let's assume we used the ON DELETE CASCADE clause on the support_rep_id in the customer table. If this was the case, and we deleted employee_id 1 from the employee table, rather than throwing an error due to the foreign key constraint, it would instead attempt to delete all of the customer records that had the support_rep_id. Subsequently, if we used ON DELETE CASCADE on all of the foreign keys of all of our tables, the database would then try to delete all the

invoices that reflected those customers that had the support_rep_id equal to 1. Then it would also delete the invoice_line records that reflect the deleted invoices.

You can quickly see why this could create a lot of potential issues in the database if we were not careful. If we wanted to add these additional clauses to the table, we would have to first drop the constraint:

```
ALTER TABLE department  
DROP CONSTRAINT fk_manager;
```

Then we could add it back on with any changes to the clauses:

```
ALTER TABLE department  
ADD CONSTRAINT fk_manager  
FOREIGN KEY (manager_id)  
REFERENCES representative (representative_id)  
ON DELETE CASCADE;
```

Video Transcription

There are different instances where you might want to add in the foreign keys after the fact, rather than when you create the tables. Such as loading in data, the process of creating different tables, you don't have to worry about the order in which you're creating the tables. So in this case here, we'll go ahead and create their representative table and the department table as a starting point. Both of them have been created. And then once we actually have those tables created, we can go ahead and add in the constraint itself by doing an ALTER TABLE statement.

And then we'll identify at the table that we're going to be altering, which is the department table, we identify adding the constraint. We're going to name the constraint fk_manager, very similar to what we've done in the prior tutorial. Identify that's going to be a foreign key on the manager ID within the department table, and then referencing the representative table with the representative ID. So you'll notice that the syntax between creating a table with the foreign key and altering the table will look very similar to one another.



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 use the ALTER TABLE statement to add a foreign key constraint to a table.

Source: Authored by Vincent Tran