

# Foreign Keys & Referential Data

by Sophia



## WHAT'S COVERED

This tutorial explores the use of foreign keys and the importance of the execution of SQL statements in two parts:

1. Inserting with Foreign Keys
2. Deleting with Foreign Keys

## 1. Inserting with Foreign Keys

When it comes to foreign keys, it can be a challenge to determine the order in which data must be inserted into tables, especially when we have multiple tables that relate to one another. Given our database, we have the following foreign keys in place across the various tables:

constraint_name	table_name	column_name	foreign_table_name	foreign_column_name
album_artist_id_fkey	album	artist_id	artist	artist_id
track_album_id_fkey	track	album_id	album	album_id
employee_reports_to_fkey	employee	reports_to	employee	employee_id
customer_support_rep_id_fkey	customer	support_rep_id	employee	employee_id
invoice_customer_id_fkey	invoice	customer_id	customer	customer_id
track_genre_id_fkey	track	genre_id	genre	genre_id
invoice_line_invoice_id_fkey	invoice_line	invoice_id	invoice	invoice_id
track_media_type_id_fkey	track	media_type_id	media_type	media_type_id
invoice_line_track_id_fkey	invoice_line	track_id	track	track_id
playlist_track_track_id_fkey	playlist_track	track_id	track	track_id
playlist_track_playlist_id_fkey	playlist_track	playlist_id	playlist	playlist_id

In looking at the list, the first column shows the name of the foreign key constraint. The second and third columns show the child table that has a foreign key. The fourth and fifth columns show the parent table and primary key that the child is referencing.

The order in which the insert statements need to occur (assuming none of the records are referencing existing records) begins with tables that do not have foreign keys to other tables. If we look at our set of table constraints, we can identify that these are the artist, genre, media\_type, playlist, and employee tables:

constraint_name	table_name	column_name	foreign_table_name	foreign_column_name
album_artist_id_fkey	album	artist_id	artist	artist_id
track_album_id_fkey	track	album_id	album	album_id
employee_reports_to_fkey	employee	reports_to	employee	employee_id
customer_support_rep_id_fkey	customer	support_rep_id	employee	employee_id
invoice_customer_id_fkey	invoice	customer_id	customer	customer_id
track_genre_id_fkey	track	genre_id	genre	genre_id
invoice_line_invoice_id_fkey	invoice_line	invoice_id	invoice	invoice_id
track_media_type_id_fkey	track	media_type_id	media_type	media_type_id
invoice_line_track_id_fkey	invoice_line	track_id	track	track_id
playlist_track_track_id_fkey	playlist_track	track_id	track	track_id
playlist_track_playlist_id_fkey	playlist_track	playlist_id	playlist	playlist_id

Note that the employee table links to itself, as the manager is linked using the reports\_to column in the same table. So, the first five tables that would need to be inserted in are:

artist  
employee  
genre  
media\_type  
playlist

The order among the five tables does not matter, as they are all on the first level of tables. Next, we look at the tables that reference those five tables.

constraint_name	table_name	column_name	foreign_table_name	foreign_column_name
album_artist_id_fkey	album	artist_id	artist	artist_id
track_album_id_fkey	track	album_id	album	album_id
employee_reports_to_fkey	employee	reports_to	employee	employee_id
customer_support_rep_id_fkey	customer	support_rep_id	employee	employee_id
invoice_customer_id_fkey	invoice	customer_id	customer	customer_id
track_genre_id_fkey	track	genre_id	genre	genre_id
invoice_line_invoice_id_fkey	invoice_line	invoice_id	invoice	invoice_id
track_media_type_id_fkey	track	media_type_id	media_type	media_type_id
invoice_line_track_id_fkey	invoice_line	track_id	track	track_id
playlist_track_track_id_fkey	playlist_track	track_id	track	track_id
playlist_track_playlist_id_fkey	playlist_track	playlist_id	playlist	playlist_id

Here we have the album, employee, customer, playlist\_track, and track table. However, if we look at the second row, we see that the track table also has a foreign key to the album\_id in the album table. The album table was not part of the first set of tables, so the track table cannot be added to yet. Likewise, in the last row, the playlist\_track has a link to the track table, which isn't available yet. So the next set of tables that can be inserted into is:

album  
customer

Next, we can identify that the following tables are linked to the ones at the prior level and do not have other dependencies:

constraint_name	table_name	column_name	foreign_table_name	foreign_column_name
album_artist_id_fkey	album	artist_id	artist	artist_id
track_album_id_fkey	track	album_id	album	album_id
employee_reports_to_fkey	employee	reports_to	employee	employee_id
customer_support_rep_id_fkey	customer	support_rep_id	employee	employee_id
invoice_customer_id_fkey	invoice	customer_id	customer	customer_id
track_genre_id_fkey	track	genre_id	genre	genre_id
invoice_line_invoice_id_fkey	invoice_line	invoice_id	invoice	invoice_id
track_media_type_id_fkey	track	media_type_id	media_type	media_type_id
invoice_line_track_id_fkey	invoice_line	track_id	track	track_id
playlist_track_track_id_fkey	playlist_track	track_id	track	track_id
playlist_track_playlist_id_fkey	playlist_track	playlist_id	playlist	playlist_id

Now, being that the album would have been inserted, the track table can be inserted into next. The invoice table only depends on the customer, so that can also be inserted into it. So, on the third level, we can insert into in any order:

constraint_name	table_name	column_name	foreign_table_name	foreign_column_name
album_artist_id_fkey	album	artist_id	artist	artist_id
track_album_id_fkey	track	album_id	album	album_id
employee_reports_to_fkey	employee	reports_to	employee	employee_id
customer_support_rep_id_fkey	customer	support_rep_id	employee	employee_id
invoice_customer_id_fkey	invoice	customer_id	customer	customer_id
track_genre_id_fkey	track	genre_id	genre	genre_id
invoice_line_invoice_id_fkey	invoice_line	invoice_id	invoice	invoice_id
track_media_type_id_fkey	track	media_type_id	media_type	media_type_id
invoice_line_track_id_fkey	invoice_line	track_id	track	track_id
playlist_track_track_id_fkey	playlist_track	track_id	track	track_id
playlist_track_playlist_id_fkey	playlist_track	playlist_id	playlist	playlist_id

Then, on the final level, we can now insert into invoice\_line and playlist\_track, as all dependencies have been accounted for.

In review, we would have to insert into any grouping of the following levels of tables:

#### Level 1

artist  
employee  
genre  
media\_type  
playlist

#### Level 2

album  
customer

### Level 3

track  
invoice

### Level 4

invoice\_line  
playlist\_track

Using this structure, if we needed to insert an invoice on an existing customer, we would only have to look at the related tables on the levels below it. In this case, it would be the invoice and invoice\_line.

---

## 2. Deleting with Foreign Keys

When it comes to deleting from tables, the order of the statements is very similar to the order of inserting, but backwards. We want to delete the items at the lowest level first and work up. This is, of course, if there is no ON DELETE CASCADE set on the foreign keys in the tables. We must delete in the opposite direction as we would insert, as we have to delete the child rows first before the parent rows.

In essence, the tables would have to be deleted in the following order by level:

### Level 4

invoice\_line  
playlist\_track

### Level 3

track  
invoice

### Level 2

album  
customer

### Level 1

artist  
employee  
genre  
media\_type  
playlist

For example, if we wanted to delete a customer, we would have to delete from the invoice\_line of the invoices that belong to that customer. Then delete the invoices in the invoice table that belong to the customer. Then, finally, delete the customer.



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

The foreign key constraints force us to insert and delete data from tables in a specific order.

Source: Authored by Vincent Tran