

# Cross Joins

by Sophia



## WHAT'S COVERED

This tutorial explores using cross join to join data in tables in two parts:

1. Introduction
2. Examples With Simple Tables

## 1. Introduction

A cross join is a unique type of join that creates a combination of every row from two different tables. It is also called a cross product or cartesian join, since all row combinations are included in the result. One example where we could use a cross join is if we wanted to get all of the potential combinations of a product, like all of its colors and sizes.

It is important to note that these cases are rare, and most scenarios where we have a cartesian join between two tables means that we have a missing join statement between a common attribute.

The structure of the cross join looks like this:

```
SELECT <columnlist>  
FROM <table1>  
CROSS JOIN <table2>;
```

Notice that there are no details of how the tables are joined. This statement would result in the same result set as if we did:

```
SELECT <columnlist>  
FROM <table1>,<table2>;
```

Or, we could even use an inner join with a condition that always evaluates to true to force a cross join:

```
SELECT <columnlist>  
FROM <table1>  
INNER JOIN <table2> ON true;
```

## 2. Examples With Simple Tables

Let's give this a try by creating a couple of tables for size and color and joining them together.

```
CREATE TABLE color ( color_id INT PRIMARY KEY, color_name VARCHAR (50) NOT NULL );
```

```
CREATE TABLE size ( size_id INT PRIMARY KEY, size_name VARCHAR (30) NOT NULL );
```

```
INSERT INTO color (color_id, color_name)
VALUES (1, 'Blue'), (2, 'Red'), (3, 'Yellow');
```

```
INSERT INTO size (size_id, size_name)
VALUES (1, 'Small'), (2, 'Medium'), (3, 'Large');
```

Now that the tables have been created, we should expect to see 9 rows returned from the cross join, as there are 3 rows in the color table and 3 in the size table:

```
SELECT color_id, color_name, size_id, size_name
FROM color
CROSS JOIN size;
```

Query Results			
Row count: 9			
color_id	color_name	size_id	size_name
1	Blue	1	Small
2	Red	1	Small
3	Yellow	1	Small
1	Blue	2	Medium
2	Red	2	Medium
3	Yellow	2	Medium
1	Blue	3	Large
2	Red	3	Large
3	Yellow	3	Large

Running each of the other queries returns the same results:

```
SELECT color_id, color_name, size_id, size_name
FROM color, size;
```

```
SELECT color_id, color_name, size_id, size_name
FROM color
INNER JOIN size ON true;
```

## Query Results

Row count: 9

color_id	color_name	size_id	size_name
1	Blue	1	Small
2	Red	1	Small
3	Yellow	1	Small
1	Blue	2	Medium
2	Red	2	Medium
3	Yellow	2	Medium
1	Blue	3	Large
2	Red	3	Large
3	Yellow	3	Large

As you can see, the cross join has taken each color and matched it to each size to display all possible combinations. This type of combination can become problematic as the size of the data gets larger. In our example, we only have 3 rows in the color table and 3 rows in the size table. But imagine if we had 1,000 rows in one table and 1,000 rows in another table. The result would have 1,000,000 rows in the result.

## Video Transcription

[MUSIC PLAYING] A cross join is a very unique type of join that allows you to be able to link every single row within the left table and joining it every single record in the right table. This is typically not utilized all that often. There's very special circumstances that you would actually utilize this. And the query itself is actually quite simple. In this case here, we have two way tables to represent of an apartment table, and we want to do is remove that joint statement. It's got to be linked to two tables and changing in our joint cross join.

And when we have that, it's going to take the data in one table and link it to all the data that's in the other table. In this case here, we have four records in the representative table and five in an apartment. So by doing so, when we run this, it'll multiply the two being that it's going to link every single word together to result into 20 different rows in this case here. And so as part of the data results, you'll be able to see that that representative ID, Bob Evans, is linked to every single row in the other table. That tango in this case here, with the value of two for representative ID matches to all the different departments, as well. And it kind of links on over and over again.

This is important to note being that if you had two different tables that had 1,000 records in each table, if you tried to do a cross join between them, you'd have 1,000 times 1,000, meaning one million records being returned in the result set.

[MUSIC PLAYING]



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 CROSS JOIN clause creates a combination of every row from two or more different tables.

Source: Authored by Vincent Tran