Steve Stedman
Freelance SQL Server Consultant
https://stevestedman.com

TSQL Basics Part 2: Inner Join

This is part 2 of a 14 part series on TSQL Basics. You will first gain an understanding of the differences between each of these types of joins, and when they should be used. Then we'll explore some common uses for joins, such as replacing a not in clause with join to get the same results with better performance. You will take home the right foundations to get at the data that you want.

Transcription:

Next, we're going to take a look at an inner join. And what an inner join does is it connects data between two tables. In this case, we just say table one and table two, you'd put your real table names there, of course, and I've aliased them as T one and T two. So that we can use the short name of T one and T two instead of the table name. And then what this does, is it returns only the rows that match on the join predicate, the join predicate is the what's people sometimes called the on clause, and this is specifying an inner join, it's also known as just a plain join, if you use the word INNER JOIN, or just the word join, it means the same thing. It's matching on those. And it's only showing you the things that are an exact match. Now, if you're watching and you're thinking, Oh, this is a little bit boring at this point, we're just getting into it. This is the simple stuff, we're going to get into the more interesting stuff here. Real quick. But the joint predicate, like I said, that's what's known as typically as the on clause, predicate, it's what specifies the relationship between those two tables. And why why is this important is that in a lot of tables that you may work with, you've got lots of different IDs. And if you don't understand what Id signifies the relationship between table one and table two, well, you may not be connecting them on the right information. I remember, years ago, I was working with a report developer, who didn't understand the naming of the different keys between the table and ended up joining on the wrong key, which lead to just completely bogus data showing up in the reports that were built off of that. So that's why I always want to talk about here is understanding the relationship. And if you don't understand the data, and how the tables work well together before you try and write your query, your query is not going to probably work very well. So we'll flip back over to Management Studio here, we've looked at our three tables, table. One is the people table two is the favorite colors. And table three is the favorite foods, probably everybody has a much more exciting database than that. But let's take a look here. And we're going to take a look at inner join. And when we run this, we're going to be joining table one and table two. And we're joining the favorite color from table one over to the ID from table two. And when we run that, you can see that the first columns here, those all came from table one, and

then the ID and the favorite color over here came from table two. And you can see that there's two people here whose favorite color was one, which was Steve and Fred. And in that example, they actually had two favorite colors, because there's an error in the data. And we all know that in real databases, you get mistakes made. But I wanted to use this for the example where there's two colors mapped to the idea of one of red and Indigo. And that shows that because there's two rows with an ID of one, in table two, it ends up duplicating out these rows when it does an inner join. So some people look at and they say, well, I've got exactly n rows that are a match, why am I seeing more than n being the number of rows showing up in the result set. And sometimes they'll match on multiple things like the red and Indigo being shown there on the join. Now, what you'll also notice is there's a couple people who got left off from that original list, if we scroll back up and take a look at table one, we can see there's Johnny and Karen down here at the bottom who don't have a favorite color, their favorite color is no, well, they got missed when we did the inner join. Because they don't there's there's no way to link them up there based off of that with an inner join. So here it's just your basic SELECT statement, you use the word inner join. And then the alternate syntax is just to use the word join rather than inner. Now I like to use the word INNER JOIN because it's explicit, it makes it so that the next person reading it is aware that you intended to do an inner join here. Instead of thinking you were doing some other type of join and you just might have forgot the keyword. So when we run this just the with the join rather than the word INNER JOIN there, we get the exact same result set that we saw with the the inner join above. Inner Joins are by far the most common join type that you'll probably do in SQL Server. Also, just a reminder, you can download the join types poster at SteveStedman.com And this is a poster I created a few years ago that just sort of shows how all the different joins that we're going to be covering through this presentation are done. It's handy to pronounce and keep on your cue ball or near your workspace.