Steve's Introduction to Recursive Common Table Expressions

Transcription of Video:

Now on to recursive common table expressions. This is my favorite part. And this is where it gets really fun is working with recursive CTE's. Because with recursive CTE, we're doing things that are really difficult to do any other way with SQL Server. So, a CTE is considered recursive when the CTE references itself. It's interesting, you can't do that in a view or have a view references itself. But in the CTE, you can actually declare the CTE and inside of that query definition, reference that same CTE itself. Now, the recursion will keep going until it hits one of the following stop criteria. One is when the query produces no results, or referencing itself and going recursively doesn't give you anything new to add into your results, or when the max recursion is hit. And the default on max recursion is 100. So if you're going 100 levels deep, or less than 100, levels deep, that's a good default there. But if you have a query that you need to go more than 100 levels deep, you need to specify different Max recursion value. Now, I think the max you can specify as like 32,000, or somewhere around there. Or you can specify zero, which implies no Max recursion, which just says keep recursing until the query either produces no results, or until the SQL Server runs out of memory, one or the other. And yes, I have run SQL server out of memory on some pretty big systems, experimenting around with recursive CTE's. Some of the uses for recursive CTE's are hierarchical listing of categories. So we looked at our outdoor store, which had snowboard sports and ski equipment, and camping and all those kinds of things. And we were able to display a list of categories. But we weren't able to do anything to really put that into a hierarchy of what's a subcategory of another. They're great for recursive calculations, things like a Fibonacci sequence. Really fun to deal with the CTE. You can also do things like kind of a tree view of like a family tree structure, or things like that, and much, much more. But in my, from my perspective, recursive CTE's are the most interesting part of common table expressions. So let's take a look at a recursion overview. If you were assigned the task of summing the numbers from one to 10, without recursion, you might put something together like this where you say 10 plus nine plus eight plus seven plus six plus five plus 4321 equals 55. That works great if you've got a single function or single math function you want to work with there. But then, if you wanted to, say sum the numbers recursively, from one to 10, you could say, well, the sum of the numbers from one to 10 just happens to be 10 plus the sum of the numbers from one to nine. And then which is 10 plus nine plus the sum of the numbers one to eight. And sort of working through the logic. Finally, we get back to the same, almost the same layout it was above is 10 plus nine plus eight plus seven plus six, but it's easier just to declare it and say it's 10 plus the sum of numbers one to nine. Now,

imagine if you're trying to do these sums for bigger numbers in the 1000s. This would take you a whole lot of time, but doing it recursively allows you to accommodate multiple depths that you're dealing with there.