Recursive CTE Examples

Transcription of Video:

One of the things I like play with is performance and pushing things to their limits. So what I want to do next is see how far can we push this sum of parts? Can we do the sum of parts for 40,000? With the option of Max recursion equaling 40,000? No, we cannot do that because the value of 40,000 specified exceeds the limit of 32,767. Okay, well, the way we can get around that, and this is the dangerous way of doing it. And we'll just copy that original line and comment it out. We say option Max recursion of 00 means go indefinitely. We run that. And now the sum of parts of 40,000 is this big, big number here with lots and lots of zeros after it. The recursion on that went 40,000 levels deep, and it stopped. So how can we prove that? Well, if we just say SELECT star and comment out the max, we run that same query. Our result set you can see down in the bottom corner here is 40,000 rows, and we scroll down to the bottom. And there's 40,000 rows that were taken out of that recursive query. Let's take a look at another table. And we're gonna do some recursion. Here, we have this royalty table. And this is just looking up some of the British Royal Family on disclaimer, I'm not related to any of them. And then we're going to look on each one. Some of them have their mother and father defined and some of them don't. Depending on how far back like anyone's family tree eventually go far enough back and you don't have who the mother and father are. So if we look down here we have queen, Queen Elizabeth, the second of Windsor. Her mother is ID of six and her father's ID of three. So here's ID of six, which is Elizabeth, Angela, Marguerite, Bowen, Lyon. And here's three, which is her father, which is George, the sixth Windsor, King of England, if I'm doing my Roman numerals correctly. But this isn't really a very good way to show a family tree. It's hard to follow. So we're going to do is we're going to build a royal family tree with a CTE, we're going to use the anchor query, and we're just going to start with ID of 17. Because that happens to be William Arthur, Philip Windsor, or Prince William as I believe he's known as when he use that as our anchor query. That's our starting point. And then we're going to say, give me for this individual ID of 17 on the recursive query, give me all of his mother or give me his mother, and all of his mother's mother's. And when we SELECT from this recursive CTE, we now see Prince William, his mother is 1616 is Diana Francis or Lady die, which her mother's 18th, which is Frances Ruth Burke, Roche, hopefully I'm pronouncing all those correctly. If I'm not, it doesn't matter for the recursion part. But we don't have the father's side showing up very well here. So what we're going to take a look at next is if we took that same query above, and we simply replace the joint of the the father, or sorry, the joint of the mother to replace it with the Father, we run this here's Prince William, his father, Prince Charles, his father, Philip, Duke of Edinburgh, Andrius Prince of Greece, and William George, the first of Helens get I'm not sure if I'm pronouncing those correctly. So we were able to track on the father's side, back. So this is like a left handed family tree and a right

handed family tree going back on one side or the other. But we're not going back on both sides. So how do we go back on both sides? Well, what we do is we take that same CTE. And here, the part that's highlighted as the exact part above that was showing the Father, we throw in a second recursive query with another union, all that says for everybody returned from from all these results, we want to also look at who the mother is. And we're going to grab that from the World Tree CTE. And that's kind of like saying select star from the original table because it gave us everybody back. But it was able to give the relationships they're starting with principally, and we've got his mother and his father and everybody on both sides.

Here, we're going to use the Fibonacci sequence. And if you're not familiar with the Fibonacci sequence, the Fibonacci sequence is basically the numbers zero and one, and then each subsequent number and I need to put a carriage return in here so we can see each subsequent number is the sum of the previous two. So zero and one is one. So then the Fibonacci sequence that would be one, one and one is two. One and two is three. Two and three is five. Three and five is eight. and so on, but it gets a little bit hard to follow in there. So let's take a look at how to do this. So we build the initial Fibonacci sequences, just select zero and one, because by definition, the Fibonacci sequence starts with zero and one. So there's the initial Fibonacci sequence zero and one, and the total value is one. But now what we're going to do is we're going to say select zero and one, and then give me an n plus the previous value plus n as the Fibonacci number, where n is less than this ginormous number right there. So we can see the Fibonacci sequence goes like this 01123, which is two plus one, five, which is three plus two, eight, which is five plus three above it 13, which is the two above 21, which is the two above 34, which is two above, and then things start getting big, really fast. And next thing, you know, we're down into some really big numbers down here, when it's taking this value in that value, and adding them up to the next one. But there you go Fibonacci sequence calculated using a common table expression, we can clean it up a little bit by in the output just displaying the number we want, rather than two columns. That's pretty straightforward there. And again, we're using the max recursion is zero just to be safe there. Or maybe not to be safe, maybe be a little bit risky, because you never know what's going to run out.

Then what we're going to do is let's try going farther. Let's try. So then what we're going to do is try building the Fibonacci sequence out into a string of text. And let's take a look at what that looks like. So here we have the Fibonacci sequence basically pivoted or kind of turned around as a single string. And I'm just going to copy this value and paste it into a new window. And you can see the Fibonacci sequence gets pretty big quickly there. But we've got the entire entire sequence put out as a VAR char value, rather than as a numeric value, showing the entire sequence. Again, just using some fun math here to kind of show off how specific things work with recursion.

Okay, let's take a look at factorial. Calculating factorials with a CTE. Well, the way a factorial works, and a factorial is represented as an exclamation point, factorial works, if five factorial is five, times four times three times two times one, which gives a value of 120. Okay, that's not recursive. And let's just start building an anchor query. So factorial of one is simple. It's one times all the numbers less than one and greater than zero, which is just one. So we select the N and the factorial, and we get the numbers one and the factorial is one. Anchor queries are usually boring, that's okay. But what we're going to do now is take that anchor query and build it into a recursive query. So here we have the declaration, the CTE, the anchor, query, the union all and then we're going to say for factorial, we're going to grab n plus one to each level D, we're going to add one to it. And then we're gonna take that n plus one value times the previous factorial, so level two, it would be two times one, or to a level three, it would be three times two, or six, and so on, we select that from factorial, and we're gonna abort when we get to five levels deep, because we don't want this to run wild and hit our recursion level. So let's run this. And we can see that factorial of one is one factorial of two is two, factorial three is six, factorial four is 24, and five is 120. Let's push it a little bit further and see what happens. Let's look for a factorial of 20. So all I've really done is just changed the five above to a 20, as it's shown here, we run it. And we get an arithmetic overflow converting expressions to data type int. Well, the data type int only holds so big. So so large of a piece of data. So it's about 4 billion ish if I remember the number off the top my head. So what we're doing there is we're getting the point to where it exceeds what can be held in it. So what we do, what we can do to fix that and get around that is we simply cast our starting value as a big end. And then everything in that column from levels on down will be a big event, which will allow for larger numbers. And you can see now we get to the factorial level of 20. Well, beyond that 4 billion limit, we're hitting with a really, really big number here. But how far can we go? Well, if we change this 20 to let's say, 30. Can we accommodate that? Well, even a big answer overflows at a factorial of 30. And in fact, did 20 was all we got to 21 was where it overflowed. But I really want to calculate a higher factorial number. So what I'm going to do do is I'm going to cast that initial starting number as a numeric 38, rather than a big N, which should let us go larger than what the big N would hold. We run that our factorial calculation, as you can see below, gets really big fast. And there we go. Factorial 33 Is this gigantic number you can see there, we try going to 34. Well, I think at this point, that's where we overflow the numeric value. Yep, same issue. So we hit the limits of the data types pretty darn quick when we're doing things like factorial. Okay, so now we're going to come back to the World Tree CTE that we're looking at. And here's the same query we looked at earlier, where we're selecting the mother and the father on both sides of the tree for number 17, which is Prince William. And there it is, but that's kind of hard to follow.

So let's see if we can figure out how to better represent it. So what we're gonna do is we're going to tag this top level person as the source, everybody on the father's side, we're going to put the word father there for the relationship, and everyone on the mother's side when you use the word mother. Now at least we can see that here is the starting person, and then everyone below that is laid out as who's a father or a mother. But it's not really clear. Like, if I was looking at this as a family tree,

it'd be really hard to follow. But we at least know who all the fathers and mothers are.

But next, what I want to do is I want to add a new column, and I'm going to call that tree path. And what tree path is, is, it's going to be a tool that we're going to use to kind of indent and lay things out a little bit so we can get a better understanding of the data. And so initially, we're just going to cast the name as a tree path. And then we're going to take the previous tree path. And on the father's side, we tack on this little arrow and father, and we put the father's name, and on the mother's side, we tack on a little arrow and mother and we put the mother's name, we're only gonna go three levels deep just to see this little bit easier to follow and take a look at. And looking here because things get wide real quick. We can see, at any level, the relationship between Prince Windsor, sorry, Prince William, is that his mother is Lady die. And her mother is Frances Ruth Burke. Or if we want to look at the relationship between Prince William and Elizabeth the Second, it's Prince William, his father is Prince Charles. And then the mother is Queen Elizabeth the Second, visualizing the data in the CTE can be as valuable as just being able to calculate it. To me, this is far more valuable than what we were looking at above where it just showed the list of who's who and who their parents are. All right, so let's take a look at doing this a different way. What we're doing now, and I wanted to bring up the results while we look at the query. So you can see them both at the same time, is that with our anchor query, we're grabbing ID of 17, which is principal M. And we're just declaring that he is the root value in the tree, the tree root. So here's the root. And then what we're going to do, as we do each level of recursion, we're going to use the previous tree path. Based off the recursion, we're going to tag each one as a father or the mother. And then down in the results, we're going to use this replicate function. And we're basically gonna say replicate a period and some spaces, level number of times. So at the root level, which is level zero, we don't get that anytime. And from there on down, we sort we indent based off of the level. And then we order by the tree path, and that ends us ends up giving us the structure here, where the tree path is what we saw in the previous query with the full name list, here we're getting an indentation that shows at the root it's real quick to see Prince William's at the root. Here's his father. Here's his mother. And then for Lady die, here's her mother and father for Prince William here's his father. Oh, sorry for Prince Charles. Here's his father Philip and Elizabeth. And for Elizabeth, her father is George and Elizabeth. And for George the fourth is sorry, George the sixth his father's George the fifth and Mary Princess of tech. There we go. Much better format and out of recursive CTE. This is this is the fun stuff.