Generating a Tree Path with a CTE

The following example shows how to create a Tree Path with a Common Table Expression (**CTE**).

First off what is a tree path?

For this example I will be referring to the product categories at a camping and fitness store, where you have a top level with 4 categories, and some of the categories have subcategories:

- Camping
    - Tents
        - 1 Person
        - 2 Person
            - Family Camping
            - Backpacking
            - Mountaineering
        - 3 Person
        - 4 Person
    - Backpacks
    - Sleeping Bags
    - Cooking
- Cycle
- Snowsports
- Fitness

In this example, the tree path for 2 person mountaineering tents would be "Camping -> Tents -> 2 Person -> Mountaineering". This is sometimes referred to as bread crumbs when you are creating a website or storefront.

So how do you get from the following table structure into a hierarchy that includes the Tree Path.

You could do it by doing a self JOIN from the table back to itself, joining parent to id, which would work for 2 levels. Then to go to the third or fourth level, you would need to another one or two self JOINs to get those levels in. The problem with this strategy is that you are limited on the level of categories or the depth of the tree.

Using a Common Table Expression

The way that I would do it is with a CTE to get the following results.

Here is how you do it, with a recursive CTE, which will accommodate any number of levels up to 100 (even more if you specify a deeper MAXRECURSION).

```
WITH departmentcte(deptid, department, parent, LEVEL, treepath) AS

( SELECT id AS deptid, department, parent, 0 AS LEVEL,

CAST(department AS VARCHAR(1024)) AS treepath

FROM departments

WHERE parent IS NULL

UNION ALL -- and now for the recursive part
SELECT d.id AS deptid, d.department, d.parent,

departmentcte.LEVEL + 1 AS LEVEL,

CAST(departmentcte.treepath + ' -> ' +

CAST(d.department AS VARCHAR(1024))

AS VARCHAR(1024)) AS treepath

FROM departments d

INNER JOIN departmentcte

ON departmentcte.deptid = d.parent)

SELECT *

FROM departmentcte

ORDER BY treepath;
```

Just another example of using a Recursive Common Table Expression.

For more details, see my CTE presentation at SQL Saturday in Redmond.