<https://docs.microsoft.com/en-us/sql/relational-databases/tables/lesson-2-4-reordering-data-in-a-hierarchical-table-using-hierarchical-methods>

# Lesson 2-4 - Reordering Data in a Hierarchical Table Using Hierarchical Methods

2017-3-6 3 min to read Contributors

* [BYHAM](https://github.com/BYHAM)

* [Craig Guyer](https://github.com/craigg-msft)

Reorganizing a hierarchy is a common maintenance task. In this task, we will use an UPDATE statement with the [GetReparentedValue](https://docs.microsoft.com/en-us/sql/t-sql/data-types/getreparentedvalue-database-engine) method to first move a single row to a new location in the hierarchy. Then we will move an entire sub-tree to a new location.

The GetReparentedValue method takes two arguments. The first argument describes the part of the hierarchy to be modified. For example, if a hierarchy is **/1/4/2/3/** and you want to change the **/1/4/** section, the hierarchy becomes **/2/1/2/3/**, leaving the last two nodes (**2/3/**) unchanged, you must provide the changing nodes (**/1/4/**) as the first argument. The second argument provides the new hierarchy level, in our example **/2/1/**. The two arguments do not have to contain the same number of levels.

### To move a single row to a new location in the hierarchy

1. Currently Wanida reports to Sariya. In this procedure, you move Wanida from her current node **/1/1/,** so that she reports to Jill. Her new node will become **/3/1/** so **/1/** is the first argument and **/3/** is the second. These correspond to the **OrgNode** values of Sariya and Jill. Execute the following code to move Wanida from Sariya's organization to Jill's:

Copy

DECLARE @CurrentEmployee hierarchyid , @OldParent hierarchyid, @NewParent hierarchyid

SELECT @CurrentEmployee = OrgNode FROM HumanResources.EmployeeOrg

WHERE EmployeeID = 269 ;

SELECT @OldParent = OrgNode FROM HumanResources.EmployeeOrg

WHERE EmployeeID = 46 ;

SELECT @NewParent = OrgNode FROM HumanResources.EmployeeOrg

WHERE EmployeeID = 119 ;

UPDATE HumanResources.EmployeeOrg

SET OrgNode = @CurrentEmployee. GetReparentedValue(@OldParent, @NewParent)

WHERE OrgNode = @CurrentEmployee ;

GO

1. Execute the following code to see the result:

Copy

SELECT OrgNode.ToString() AS Text\_OrgNode,

OrgNode, OrgLevel, EmployeeID, EmpName, Title

FROM HumanResources.EmployeeOrg ;

GO

Wanida is now at node **/3/1/**.

### To reorganize a section of a hierarchy

1. To demonstrate how to move a larger number of people at the same time, first execute the following code to add an intern reporting to Wanida:

Copy

EXEC AddEmp 269, 291, 'Kevin', 'Marketing Intern' ;

GO

1. Now Kevin reports to Wanida, who reports to Jill, who reports to David. That means that Kevin is at level **/3/1/1/**. To move all of Jill's subordinates to a new manager, we will update all nodes that have **/3/** as their **OrgNode** to a new value. Execute the following code to update Wanida to report to Sariya, but keep Kevin reporting to Wanida:

Copy

DECLARE @OldParent hierarchyid, @NewParent hierarchyid

SELECT @OldParent = OrgNode FROM HumanResources.EmployeeOrg

WHERE EmployeeID = 119 ; -- Jill

SELECT @NewParent = OrgNode FROM HumanResources.EmployeeOrg

WHERE EmployeeID = 46 ; -- Sariya

DECLARE children\_cursor CURSOR FOR

SELECT OrgNode FROM HumanResources.EmployeeOrg

WHERE OrgNode.GetAncestor(1) = @OldParent;

DECLARE @ChildId hierarchyid;

OPEN children\_cursor

FETCH NEXT FROM children\_cursor INTO @ChildId;

WHILE @@FETCH\_STATUS = 0

BEGIN

START:

DECLARE @NewId hierarchyid;

SELECT @NewId = @NewParent.GetDescendant(MAX(OrgNode), NULL)

FROM HumanResources.EmployeeOrg WHERE OrgNode.GetAncestor(1) = @NewParent;

UPDATE HumanResources.EmployeeOrg

SET OrgNode = OrgNode.GetReparentedValue(@ChildId, @NewId)

WHERE OrgNode.IsDescendantOf(@ChildId) = 1;

IF @@error <> 0 GOTO START -- On error, retry

FETCH NEXT FROM children\_cursor INTO @ChildId;

END

CLOSE children\_cursor;

DEALLOCATE children\_cursor;

1. Execute the following code to see the result:

Copy

SELECT OrgNode.ToString() AS Text\_OrgNode,

OrgNode, OrgLevel, EmployeeID, EmpName, Title

FROM HumanResources.EmployeeOrg ;

GO

Here is the result set.

Copy

Text\_OrgNode OrgNode OrgLevel EmployeeID EmpName Title

------------ ------- -------- ---------- ------- -----------------

/ Ox 0 6 David Marketing Manager

/1/ 0x58 1 46 Sariya Marketing Specialist

/1/1/ 0x5AC0 2 269 Wanida Marketing Assistant

/1/1//2 0x5AD0 3 291 Kevin Marketing Intern

/2/ 0x68 1 271 John Marketing Specialist

/2/1/ 0x6AC0 2 272 Mary Marketing Assistant

/3/ 0x78 1 119 Jill Marketing Specialist

The entire organizational tree that had reported to Jill (both Wanida and Kevin) now reports to Sariya.

For a stored procedure to reorganize a section of a hierarchy, see the "Moving Subtrees" section of [Moving Subtrees](https://docs.microsoft.com/en-us/sql/relational-databases/hierarchical-data-sql-server#BKMK_MovingSubtrees).

## Next Task in Lesson

[Summary: Managing Data in a Hierarchical Table](https://docs.microsoft.com/en-us/sql/relational-databases/tables/lesson-2-5-summary-managing-data-in-a-hierarchical-table)