<https://docs.microsoft.com/en-us/sql/relational-databases/tables/lesson-2-3-querying-a-hierarchical-table-using-hierarchy-methods>

# Lesson 2-3 - Querying a Hierarchical Table Using Hierarchy Methods

2017-3-6 1 min to read Contributors

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

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

Now that the HumanResources.EmployeeOrg table is fully populated, this task will show you how to query the hierarchy using some of the hierarchical methods.

### To find subordinate nodes

1. Sariya has one subordinate employee. To query for Sariya's subordinates, execute the following query that uses the [IsDescendantOf](https://docs.microsoft.com/en-us/sql/t-sql/data-types/isdescendantof-database-engine) method:

Copy

DECLARE @CurrentEmployee hierarchyid

SELECT @CurrentEmployee = OrgNode

FROM HumanResources.EmployeeOrg

WHERE EmployeeID = 46 ;

SELECT \*

FROM HumanResources.EmployeeOrg

WHERE OrgNode.IsDescendantOf(@CurrentEmployee ) = 1 ;

The result lists both Sariya and Wanida. Sariya is listed because she is the descendant at the 0 level. Wanida is the descendant at the 1 level.

1. You can also query for this information by using the [GetAncestor](https://docs.microsoft.com/en-us/sql/t-sql/data-types/getancestor-database-engine) method. GetAncestor takes an argument for the level that you are trying to return. Since Wanida is one level underneath Sariya, use GetAncestor(1) as demonstrated in the following code:

Copy

DECLARE @CurrentEmployee hierarchyid

SELECT @CurrentEmployee = OrgNode

FROM HumanResources.EmployeeOrg

WHERE EmployeeID = 46 ;

SELECT OrgNode.ToString() AS Text\_OrgNode, \*

FROM HumanResources.EmployeeOrg

WHERE OrgNode.GetAncestor(1) = @CurrentEmployee

This time the result lists only Wanida.

1. Now change the @CurrentEmployee to David (EmployeeID 6) and the level to 2. Execute the following to also return Wanida:

Copy

DECLARE @CurrentEmployee hierarchyid

SELECT @CurrentEmployee = OrgNode

FROM HumanResources.EmployeeOrg

WHERE EmployeeID = 6 ;

SELECT OrgNode.ToString() AS Text\_OrgNode, \*

FROM HumanResources.EmployeeOrg

WHERE OrgNode.GetAncestor(2) = @CurrentEmployee

This time, you also receive Mary who also reports to David, two levels down.

### To use GetRoot, and GetLevel

1. As the hierarchy grows larger it is more difficult to determine where the members are in the hierarchy. Use the [GetLevel](https://docs.microsoft.com/en-us/sql/t-sql/data-types/getlevel-database-engine) method to find how many levels down each row is in the hierarchy. Execute the following code to view the levels of all the rows:

Copy

SELECT OrgNode.ToString() AS Text\_OrgNode,

OrgNode.GetLevel() AS EmpLevel, \*

FROM HumanResources.EmployeeOrg ;

GO

1. Use the [GetRoot](https://docs.microsoft.com/en-us/sql/t-sql/data-types/getroot-database-engine) method to find the root node in the hierarchy. The following code returns the single row which is the root:

Copy

SELECT OrgNode.ToString() AS Text\_OrgNode, \*

FROM HumanResources.EmployeeOrg

WHERE OrgNode = hierarchyid::GetRoot() ;

GO

The next task will reorganize the hierarchy.

## Next Task in Lesson

[Reordering Data in a Hierarchical Table Using Hierarchical Methods](https://docs.microsoft.com/en-us/sql/relational-databases/tables/lesson-2-4-reordering-data-in-a-hierarchical-table-using-hierarchical-methods)