Hierarchy

Answer 1:  
-- update the first employee as manager  
 DECLARE @Manager hierarchyid;   
SET @Manager = CAST('/' AS hierarchyid);

update [HumanResources].[Employee]

set [OrganizationNode] = @Manager

where [BusinessEntityID] = 1

-- Create a UDF to return the full display path of a node

CREATE FUNCTION dbo.fnGetFullDisplayPath(@EmployeeNodeId hierarchyid) RETURNS varchar(max)

AS

BEGIN

-- Start with the specified node

DECLARE @Depth smallint

DECLARE @DisplayPath varchar(max)

SELECT @Depth = OrganizationNode.GetLevel(), @DisplayPath = ([FirstName] + ' ' + [LastName])

FROM [HumanResources].[Employee] join [Person].[Person]

on [HumanResources].[Employee].BusinessEntityID = [Person].[Person].BusinessEntityID

WHERE OrganizationNode = @EmployeeNodeId

-- Loop through all its ancestors

DECLARE @LevelCounter smallint = 0

WHILE @LevelCounter < @Depth BEGIN

SET @LevelCounter += 1

-- Get parent node ID

DECLARE @ParentEmployeeNodeId hierarchyid =

(SELECT OrganizationNode.GetAncestor(@LevelCounter) FROM [HumanResources].[Employee]

WHERE OrganizationNode = @EmployeeNodeId)

-- Get parent name

DECLARE @ParentEmployeeName varchar(max) =

(SELECT [FirstName] + ' ' + [LastName]

FROM [HumanResources].[Employee] join [Person].[Person]

on [HumanResources].[Employee].BusinessEntityID = [Person].[Person].BusinessEntityID

WHERE OrganizationNode = @ParentEmployeeNodeId)

-- Prepend to display path

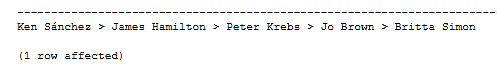
SET @DisplayPath = @ParentEmployeeName + ' > ' + @DisplayPath

END

RETURN(@DisplayPath)

END

GO

-- exec the function  
select dbo.fnGetFullDisplayPath(0x7AD6F0) 

Answer 2:

A:  
DECLARE @terriodeId hierarchyid = (SELECT [OrganizationNode]   
FROM [HumanResources].[Employee] join [Person].[Person]  
on [HumanResources].[Employee].BusinessEntityID = [Person].[Person].BusinessEntityID  
WHERE [FirstName] = 'Terri' and [LastName] = 'Duffy')  
  
SELECT \*, [OrganizationNode].ToString() AS NodeIdPath, dbo.fnGetFullDisplayPath([OrganizationNode]) AS NodeIdDisplayPath  
 FROM [HumanResources].[Employee]  
 WHERE [OrganizationNode].GetAncestor(1) = @terriodeId  
 ORDER BY NodeIdDisplayPath  
  
GO



B:  
DECLARE @AmyNodeId hierarchyid = (SELECT [OrganizationNode]

FROM [HumanResources].[Employee]

WHERE [BusinessEntityID] =1)

declare @last\_in\_hierarchyid int

select @last\_in\_hierarchyid = (SELECT top (1) OrganizationLevel

FROM [HumanResources].[Employee]

WHERE [OrganizationNode].IsDescendantOf(@AmyNodeId) = 1

ORDER BY OrganizationLevel desc)

SELECT [FirstName] + ' ' + [LastName] as name ,OrganizationNode,

OrganizationLevel,[OrganizationNode].ToString() AS NodeIdPath,

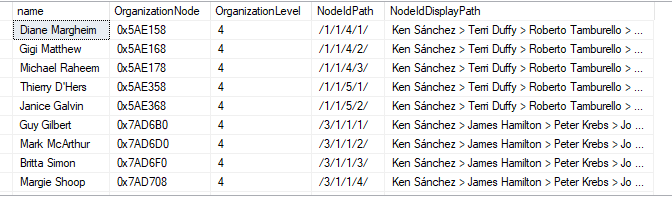
dbo.fnGetFullDisplayPath([OrganizationNode]) AS NodeIdDisplayPath

FROM [HumanResources].[Employee] join [Person].[Person]

on [HumanResources].[Employee].BusinessEntityID = [Person].[Person].BusinessEntityID

WHERE OrganizationLevel = @last\_in\_hierarchyid

GO



C:  
SELECT [HumanResources].[Employee].BusinessEntityID, [FirstName] + ' ' + [LastName], [OrganizationNode].ToString() AS NodeIdPath,

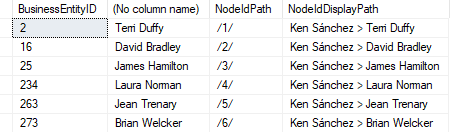
dbo.fnGetFullDisplayPath([OrganizationNode]) AS NodeIdDisplayPath

FROM [HumanResources].[Employee] join [Person].[Person]

on [HumanResources].[Employee].BusinessEntityID = [Person].[Person].BusinessEntityID

WHERE [OrganizationNode].GetLevel() = 1

GO



D:  
-- retrieve a subtree beginning with brian Welcker

DECLARE @brianNodeId hierarchyid = (SELECT [OrganizationNode]

FROM [HumanResources].[Employee] join [Person].[Person]

on [HumanResources].[Employee].BusinessEntityID = [Person].[Person].BusinessEntityID

WHERE [FirstName]= 'brian' and [LastName]= 'Welcker' )

SELECT [FirstName] + ' ' + [LastName], [OrganizationNode].ToString() AS NodeIdPath,

dbo.fnGetFullDisplayPath([OrganizationNode]) AS NodeIdDisplayPath

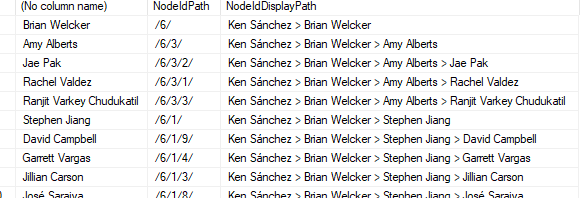
FROM [HumanResources].[Employee] join [Person].[Person]

on [HumanResources].[Employee].BusinessEntityID = [Person].[Person].BusinessEntityID

WHERE [OrganizationNode].IsDescendantOf(@brianNodeId) = 1

ORDER BY NodeIdDisplayPath

GO



E:  
-- retrieve a subtree beginning with Peter Krebs   
DECLARE @brianNodeId hierarchyid = (SELECT [OrganizationNode]

FROM [HumanResources].[Employee] join [Person].[Person]

on [HumanResources].[Employee].BusinessEntityID = [Person].[Person].BusinessEntityID

WHERE [FirstName]= 'Peter' and [LastName]= 'Krebs' )

SELECT [FirstName] + ' ' + [LastName], [OrganizationNode].ToString() AS NodeIdPath,

dbo.fnGetFullDisplayPath([OrganizationNode]) AS NodeIdDisplayPath

FROM [HumanResources].[Employee] join [Person].[Person]

on [HumanResources].[Employee].BusinessEntityID = [Person].[Person].BusinessEntityID

WHERE [OrganizationNode].IsDescendantOf(@brianNodeId) = 1

ORDER BY NodeIdDisplayPath

GO



F:

declare @GailEmployeeId hierarchyid = (SELECT OrganizationNode

FROM [HumanResources].[Employee] emp join [Person].[Person] pp

ON emp.[BusinessEntityID] = pp.[BusinessEntityID]

where pp.FirstName = 'Gail' and pp.BusinessEntityID = '5')

declare @GailAncestor hierarchyid = (select @GailEmployeeId.GetAncestor(1))

SELECT pp.FirstName, pp.LastName,

[OrganizationNode].ToString() AS NodeIdPath,

dbo.fnGetFullDisplayPath([OrganizationNode]) AS NodeIdDisplayPath

FROM [HumanResources].[Employee] emp join [Person].[Person] pp

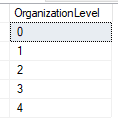
ON emp.[BusinessEntityID] = pp.[BusinessEntityID]

WHERE OrganizationNode= @GailAncestor

ORDER BY NodeIdDisplayPath



G:

DECLARE @daveNodeId hierarchyid = (SELECT [OrganizationNode]

FROM [HumanResources].[Employee]

WHERE [BusinessEntityID] = 1)

SELECT distinct OrganizationLevel

FROM [HumanResources].[Employee]

WHERE [OrganizationNode].IsDescendantOf(@daveNodeId) = 1

ORDER BY OrganizationLevel

GO  
  
Answer 3:

Declare @OrganizationNode hierarchyid, @fname varchar(20), @lname varchar(20)

declare @counter int = 0

declare @total int = ((select count(OrganizationLevel) from [HumanResources].[Employee]) - 1)

declare @levels int = (select max(OrganizationLevel) from [HumanResources].[Employee])

Declare MycursorOuter cursor

for SELECT emp.OrganizationNode, pp.FirstName, pp.LastName

FROM [HumanResources].[Employee] emp

inner join

[Person].[Person] pp

ON emp.[BusinessEntityID] = pp.[BusinessEntityID]

open MycursorOuter

Fetch next from MycursorOuter into @OrganizationNode, @fname , @lname

while @@FETCH\_STATUS=0

begin

SELECT @counter = count(\*)

FROM [HumanResources].[Employee]

WHERE OrganizationNode.IsDescendantOf(@OrganizationNode)=1

if @fname = 'Ken' and @lname = 'Sבnchez'

Print 'The number of employees under '+@fname+' '+@lname+ ' is: '+cast(@total as char(10))

else

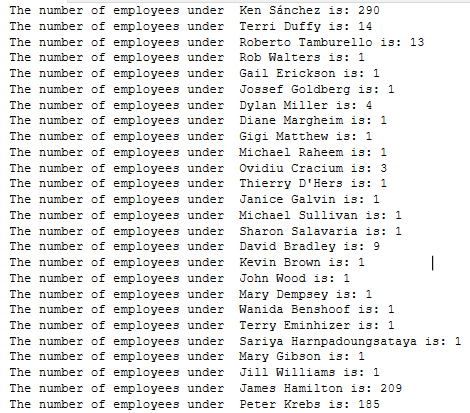
Print 'The number of employees under '+@fname+' '+@lname+ ' is: '+cast(@counter as char(10))

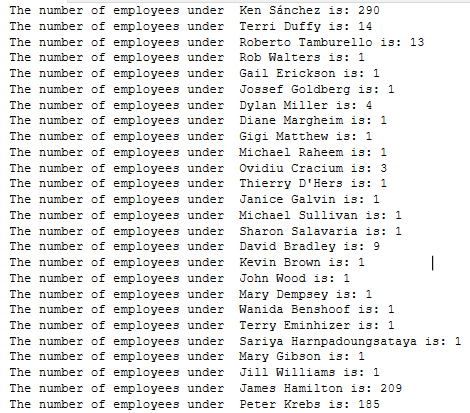
Fetch next from MycursorOuter into @OrganizationNode, @fname , @lname

end

close MycursorOuter

Deallocate MycursorOuter



Answer 4:

A. alter TABLE [HumanResources].[Employee]

add [rank] AS (OrganizationNode.GetLevel())

B. alter TABLE [HumanResources].[Employee]

add [FullDisplayPath] AS (dbo.fnGetFullDisplayPath(OrganizationNode))

C. alter TABLE [HumanResources].[Employee]

add [DirectManager] AS (OrganizationNode.GetAncestor(1))

D. alter TABLE [HumanResources].[Employee]

add [ParentDirectManager] AS (OrganizationNode.GetAncestor(2))

Answer 6:

A. DECLARE @OldParentNodeId hierarchyid = (SELECT OrganizationNode

FROM [HumanResources].[Employee] emp join [Person].[Person] pp

ON emp.[BusinessEntityID] = pp.[BusinessEntityID]

where pp.FirstName = 'Peter' and pp.LastName = 'Krebs')

DECLARE @NewParentNodeId hierarchyid = (SELECT OrganizationNode

FROM [HumanResources].[Employee] emp join [Person].[Person] pp

ON emp.[BusinessEntityID] = pp.[BusinessEntityID]

where pp.FirstName = 'David' and pp.LastName = 'Liu')

UPDATE [HumanResources].[Employee]

SET OrganizationNode = OrganizationNode.GetReparentedValue(@OldParentNodeId, @NewParentNodeId)

WHERE OrganizationNode.IsDescendantOf(@OldParentNodeId) = 1

AND OrganizationNode <> @OldParentNodeId -- Excludes Peter himself

SELECT \*, OrganizationNode.ToString() AS NodeIdPath,

dbo.fnGetFullDisplayPath(OrganizationNode) AS NodeIdDisplayPath

FROM [HumanResources].[Employee]

where [OrganizationNode] = 0x85AB58

ORDER BY NodeIdDisplayPath

DECLARE @ParentEmployeeNodeId hierarchyid =(SELECT [OrganizationNode].GetAncestor(1)

FROM [HumanResources].[Employee]

WHERE [OrganizationNode] = 0x85AB58)

SELECT pp.FirstName

FROM [HumanResources].[Employee] emp inner join [Person].[Person] pp

ON emp.[BusinessEntityID] = pp.[BusinessEntityID]

WHERE isnull([OrganizationNode],0x) = @ParentEmployeeNodeId

B. create FUNCTION dbo.fnGetRequestedDisplayPath(@ParentEmployeeNodeId hierarchyid, @Depth smallint, @LevelCounter smallint) RETURNS varchar(max)

AS

BEGIN

DECLARE @DisplayPath varchar(max)

set @ParentEmployeeNodeId = isnull(@ParentEmployeeNodeId,0x)

SELECT @DisplayPath = pp.FirstName

FROM [HumanResources].[Employee] emp join [Person].[Person] pp

ON emp.[BusinessEntityID] = pp.[BusinessEntityID]

WHERE isnull([OrganizationNode],0x) = @ParentEmployeeNodeId

set @LevelCounter = isnull(@LevelCounter,0)

WHILE @LevelCounter < @Depth

BEGIN

SET @LevelCounter += 1

DECLARE @EmployeeNodeId hierarchyid = (select @ParentEmployeeNodeId.GetDescendant(null, NULL))

DECLARE @EmployeeName varchar(max) = (SELECT pp.FirstName

FROM [HumanResources].[Employee] emp join [Person].[Person] pp

ON emp.[BusinessEntityID] = pp.[BusinessEntityID]

WHERE isnull([OrganizationNode],0x) = @EmployeeNodeId)

SET @DisplayPath = @DisplayPath+ ' > '+@EmployeeName

END

RETURN(@DisplayPath)

END