Back to Resume

BY: Christopher Singleton

SQL Scripts with Employee Manager Relationships

Note: These examples give an idea of Employee Manager relationships and how to go about finding the hierarchy information. I have provided the creating of the test database to show the output based on the queries.

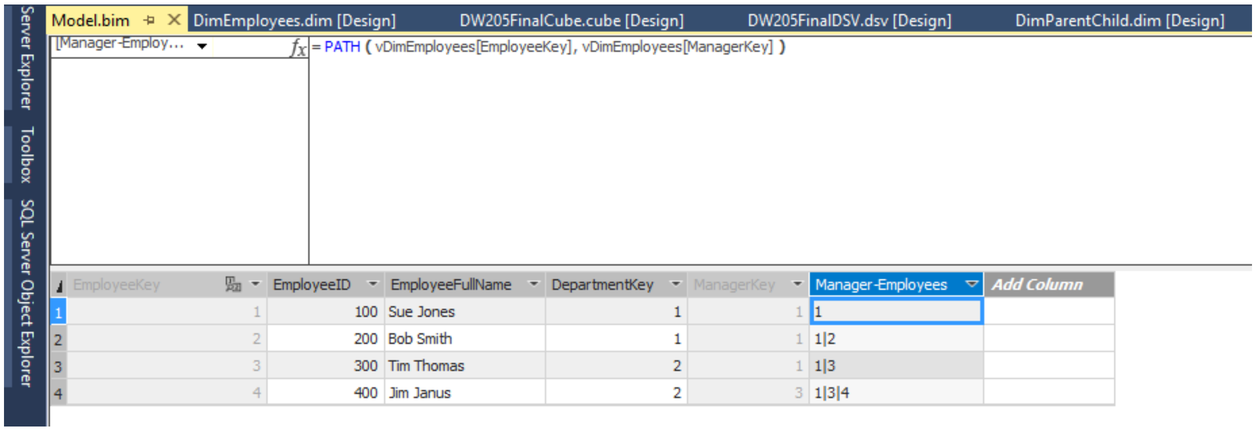
```
/*****************************************************************
****************** Employee to Manager DB Example ***********************
*************************** [WorkDB] *********************************
*****************************************************************/
USE [master];
GO

--=============================================================--
--====================== Check IF WorkDB Exists =======================--
--=============================================================--
IF EXISTS (SELECT name FROM sys.databases WHERE name = N'WorkDB')
 BEGIN
 ALTER DATABASE [WorkDB] SET SINGLE_USER WITH ROLLBACK IMMEDIATE

 --=============================================================--
--================== Drop This Database and End Function =================--
--=============================================================--
 DROP DATABASE [WorkDB] --If it already exists so we can start fresh.
/*Print out that the table was dropped,
  Convert to sysdatetime and then cast to varchar. */
 PRINT 'WorkDB Database: Dropped Database Successfully.'
 + CAST(CONVERT(varchar, SYSDATETIME(), 121) AS varchar (20))
 END

 --No Need For "GO" here.
 --============================================================--
 --==================== Create the WorkDB and use it =================--
 --============================================================--
 CREATE DATABASE WorkDB;
 GO
 USE WorkDB;

 --No Need For "GO" here.
 --============================================================--
 --================= Set the WorkDB to System Admin ===================--
 --============================================================--
EXEC [WorkDB].dbo.sp_changedbowner @loginame = N'SA', @map=false
GO
```

```sql
--==================================================================--
--================ IF Exists Drop Table, then Create the Table ==============--
--====================== [dbo].[Employee] ============================--
--Employees' ManagerID is their Manager's EmployeeID
IF OBJECT_ID('dbo.Table', 'U') IS NOT NULL
 DROP TABLE [dbo].[Employee];
GO
CREATE TABLE Employee
	(EmployeeID INT PRIMARY KEY NOT NULL
			,EmployeeName VARCHAR(25)
			,Title VARCHAR(25)
			,ManagerID INT
			,HireDate DATETIME
			,Salary INT
			,DepartmentID INT
			)

INSERT INTO Employee
	VALUES(2801, 'Ryan', 'President', NULL, '05/10/2015', 200000, 10),
			(2632, 'John', 'IT Manager', 2801, '05/01/2016', 145000, 20),
				(2755, 'Eric', 'Finance Manager', 2801, '12/01/2015', 115000, 30),
			(2600, 'David', 'Sales Manager', 2801, '07/08/2015', 110000, 40),
				(2933, 'Allen', 'BI Developer', 2632, '09/02/2017', 125000, 20),
				(2818, 'Mike', 'Data Analyst', 2632, '02/25/2016', 70000, 20),
				(2511, 'James', 'Accountant', 2755, '02/01/2015', 55000, 30),
				(2786, 'Clark', 'Accounting Assistant', 2755, '09/28/2016', 35000, 30),
				(2811, 'Bruce', 'Salesman', 2600, '05/03/2015', 40000, 40),
				(2683, 'Paul', 'Salesman', 2600, '06/03/2015', 38000, 40);


--Test:
--SELECT * FROM Employee ORDER BY departmentID

--==================================================================--
--============ Get all employees and their manager's name ================--
--==================================================================--
--Logic approach: employee's manager id =(join) manager's employee id

SELECT e.EmployeeID
     ,e.EmployeeName AS Employee
          ,e.Title
          ,e.ManagerID
          ,m.EmployeeName AS Manager
FROM Employee AS e
    LEFT JOIN Employee AS m
        ON e.ManagerID = m.EmployeeID
ORDER BY m.EmployeeName
```

```
/*Result:
EmployeeID    Employee        Title                   ManagerID          Manager
2801            Ryan          President                  NULL               NULL
2811            Bruce         Salesman                   2600               David
2683            Paul          Salesman                   2600               David
2511            James         Accountant                 2755               Eric
2786            Clark         Accounting Assistant       2755               Eric
2818            Mike          Data Analyst               2632               John
2933           Allen          BI Developer               2632               John
2600           David          Sales Manager              2801               Ryan
2632           John           IT Manager                 2801               Ryan
2755           Eric           Finance Manager            2801               Ryan
*/

--==================================================================--
--===== Get all employees who joined the company before their managers =====--
--==================================================================--
SELECT e.EmployeeID
     ,e.EmployeeName AS Employee
          ,CONVERT(VARCHAR(10), e.HireDate, 110) AS EMP_HireDate
          ,e.Title
          ,e.ManagerID
          ,m.EmployeeName AS Manager
          ,CONVERT(VARCHAR(10), m.HireDate, 110) AS MGR_HireDate
FROM Employee AS e
    LEFT JOIN Employee AS m
          ON e.ManagerID = m.EmployeeID
WHERE e.HireDate < m.HireDate

/* Result:
EmployeeID        Employee       Title          ManagerID      Manager
2511                James        Accountant        2755           Eric
2683                Paul         Salesman          2600           David
2811                Bruce        Salesman          2600           David
2818                Mike         Data Analyst       2632          John
*/

GO

--==================================================================--
--==================== Give Info on Employee Table =====================--
--========================= [dbo].[Employee] =========================--
--Note: Can be useful for checking that the table was created correctly.
EXEC sp_help [Employee];
```

<u>Using DAX in Tabular Mode to Show Employee Manager Relationships</u>

<u>Note</u>: Below is an idea of how to show Employee Manager relationship using DAX in Tabular mode inside Visual Studio while using Views as an effective over lay that does not change the original database structure, though allows you to manipulate the content for what is needed.



Server Explorer
Toolbox
SQL Server Object Explorer
Model.bim
DimEmployees.dim [Design]
DW205FinalCube.cube [Design]
DW205FinalDSV.dsv [Design]
DimParentChild.dim [Design]
[Manager-Employ…
Add Column

```
= PATH ( vDimEmployees[EmployeeKey], vDimEmployees[ManagerKey] )
```

	EmployeeKey	EmployeeID	EmployeeFullName	DepartmentKey	ManagerKey	Manager-Employees
1	1	100	Sue Jones	1	1	1
2	2	200	Bob Smith	1	1	1|2
3	3	300	Tim Thomas	2	1	1|3
4	4	400	Jim Janus	2	3	1|3|4

Code (Below):

```
=PATH (vDimEmployees[EmployeeKey], vDimEmployees[ManagerKey])
```