--USING RECUSIVE CTE(common table expressions)

USE [MyPractingDataBase]

GO

CREATE TABLE Cte\_EmployeesAndManagers

(

ID int identity (1,1) not null,

Names nvarchar(5) not null,

ManagersID int null

)

ALTER TABLE Cte\_EmployeesAndManagers

ALTER COLUMN Names nvarchar(10)

insert into Cte\_EmployeesAndManagers values ('Rob', 1)

select \* from Cte\_EmployeesAndManagers

sp\_rename 'Cte\_EmployeesAndManagers.ID', 'EmployeesID', 'COLUMN';

**……………………………………………………………………………………………………………………………………………………………………………………………..**

**……………………………………………………………………………………………………………………………………………………………………………………………..**

--WE WANT TO GET THE EMPLOYEES NAMES, THEIR MANAGERS NAMES AND THEIR LEVELS

with EmployeesAndManagersCTE (EmployeeID, Names, ManagersID, [level1])

AS

(

select EmployeesID, Names, ManagersID, 1

from Cte\_EmployeesAndManagers --since we want to get the level table, Sam is the one with no manager so he is the level one

where ManagersID IS NULL

union all

select Cte\_EmployeesAndManagers.EmployeesID, Cte\_EmployeesAndManagers.Names,

Cte\_EmployeesAndManagers.ManagersID,

EmployeesAndManagersCTE.[level1] + 1

from Cte\_EmployeesAndManagers

join EmployeesAndManagersCTE

ON Cte\_EmployeesAndManagers.ManagersID = EmployeesAndManagersCTE.EmployeeID

)

SELECT EName.Names as Employees, isnull( MName.Names, 'Super Boss' )as Managers, EName.[level1]

from EmployeesAndManagersCTE EName

left join EmployeesAndManagersCTE MName

On EName.ManagersID = MName.EmployeeID

--select \* from EmployeesAndManagersCTE

-- Now LET'S DISPLAY THE MANAGERS NAMES USING A SELF JOIN

