

```
/*
Author: Doug Streitenberger
Date: 01/10/2006
Purpose: This T-SQL hand coded to support Projects web environment and
         to demonstrate T-SQL capabilities.
*/
```

```
--*****
--SELECT Managers and those Employees that report to them
--RIGHT JOIN returns all employees --even those that have no manager assigned
--INNER JOIN does not return employees that have no manager
SELECT
    [Manager].[LastName] AS [ManagerLastName]
    , [Manager].[FirstName] AS [ManagerFirstName]
    , [Emp].[LastName] AS [EmpLastName]
    , [Emp].[FirstName] AS [EmpFirstName]
FROM
    [Employees] [Manager]
RIGHT JOIN [Employees] [Emp] ON
    [Manager].[EmployeeID] = [Emp].[ReportsTo]
ORDER BY
    [Manager].[LastName]
    , [Manager].[FirstName]
    , [Emp].[LastName]
    , [Emp].[FirstName]
```

```
--*****
--SELECT TOP 10 Customers from Customers Table based on total sales
SELECT TOP 10
    [CustomerID]
    , Cast(Sum([UnitPrice]
        * Cast([Quantity] AS Decimal)
        * (1.00 - [discount])) AS Decimal(12,2)) AS [SumOrders]
FROM
    [Orders]
INNER JOIN [Order Details] ON
    [Orders].[OrderID] = [Order Details].[OrderID]
GROUP BY
    [CustomerID]
ORDER BY
    [SumOrders] DESC
```

```
--*****
--SELECT Customers from Customers Table that have never ordered
SELECT
```

```

        [Customers].[CustomerID]
FROM
    [Customers]
LEFT JOIN [Orders] ON
    [Orders].[CustomerID] = [Customers].[CustomerID]
WHERE
    ([Orders].[CustomerID] IS NULL)

```

```

--*****
--SELECT TOP 10 Products based on quantity sold
SELECT TOP 10
    Sum([Quantity]) AS [SumProducts]
    , [ProductName]
FROM
    [Order Details]
INNER JOIN [Products] ON
    [Order Details].[ProductID] = [Products].[ProductID]
GROUP BY
    [Order Details].[ProductID], [ProductName]
ORDER BY
    [SumProducts] DESC

```

```

--*****
--CREATE A Temporary table to hold analysis
--CREATE A CURSOR for all customers from Customers table
--ITERATE Through the Customers dataset
--POPULATE Temporary table with customer standings
--GET results from Temporary table
CREATE TABLE #CustomerStandings
(
    CustomerID Varchar(5) NOT NULL
    , CustomerStanding Varchar(15) NOT NULL
)
GO
DECLARE @sCustomerID Varchar(5)
    , @sCustomerStanding Varchar(15)
    , @dcCustomerSales Decimal(12, 2)
DECLARE [CustCursor] CURSOR FOR
    SELECT [CustomerID] FROM [Customers]
OPEN [CustCursor]
FETCH NEXT FROM [CustCursor] INTO @sCustomerID
WHILE @@FETCH_STATUS = 0
BEGIN
    SELECT @dcCustomerSales =
        Cast(Sum([UnitPrice]

```

```

        * Cast([Quantity] AS Decimal)
        * (1.00 - [discount])) AS Decimal(12,2))
FROM
    [Orders]
INNER JOIN [Order Details] ON
    [Orders].[OrderID] = [Order Details].[OrderID]
WHERE
    ([CustomerID] = @sCustomerID)
IF (@dcCustomerSales > 25000)
    SELECT @sCustomerStanding = 'PLATINUM'
IF (@dcCustomerSales > 20000 AND @dcCustomerSales < 25000)
    SELECT @sCustomerStanding = 'GOLD'
IF (@dcCustomerSales > 10000 AND @dcCustomerSales < 20000)
    SELECT @sCustomerStanding = 'SILVER'
IF (@dcCustomerSales > 0 AND @dcCustomerSales < 10000)
    SELECT @sCustomerStanding = 'BRONZE'
IF (@dcCustomerSales = 0 OR @dcCustomerSales IS NULL)
    SELECT @sCustomerStanding = 'NONE'

INSERT INTO [#CustomerStandings]
    ([CustomerID], [CustomerStanding])
VALUES(@sCustomerID, @sCustomerStanding)

FETCH NEXT FROM [CustCursor] INTO @sCustomerID
END
CLOSE [CustCursor]
DEALLOCATE [CustCursor]
SELECT
    [#CustomerStandings].[CustomerID]
    ,[#CustomerStandings].[CustomerStanding]
    ,[Customers].[CompanyName]
FROM [#CustomerStandings]
INNER JOIN [Customers] ON
    [#CustomerStandings].[CustomerID] =[Customers].[CustomerID]
DROP TABLE [dbo].[#CustomerStandings]
GO

--*****
CREATE PROC [spCustomersDelete]
    @sCustomerID varchar(5)
AS
--DELETE from Customers Table based input parameter '@sCustomerID'
DELETE FROM
    [Customers]
WHERE
    ([CustomerID] = @sCustomerID)

```

GO

--\*\*\*\*\*

```
CREATE PROC [spCustomersEnum]
AS
--GET Customer Information from Customers Table on all Customer records
SELECT
    [CustomerID]
    , [CompanyName]
    , [ContactName]
    , [Phone]
    , [Fax]
FROM
    [Customers]
ORDER BY
    [CustomerID]
GO
```

--\*\*\*\*\*

```
CREATE PROC [spCustomersEnumByID]
@sCustomerID varchar(5)
AS
--GET Customer Information from Customers Table
--based input parameter '@sCustomerID'
SELECT
    [CustomerID]
    , [CompanyName]
    , [ContactName]
    , [ContactTitle]
    , [Address]
    , [City]
    , [Region]
    , [PostalCode]
    , [Country]
    , [Phone]
    , [Fax]
    , [Last_Updated]
FROM
    [Customers]
WHERE
    ([CustomerID] = @sCustomerID)
GO
```

--\*\*\*\*\*

```

CREATE PROC [spCustomersEnumCustomerIDs]
AS
--GET CustomerIDs from Customers Table on all Customer records
SELECT
    [CustomerID]
FROM
    [Customers]
ORDER BY
    [CustomerID]
GO

--*****
CREATE PROC [spCustomersEnumSearch]
    @sCustomerID varchar(6)
    ,@sCompanyName varchar(40)
    ,@sContactName varchar(30)
AS
--GET Customer Information from Customers Table based upon input
--parameters '@sCustomerID', '@sCompanyName', '@sContactName'
SELECT
    [CustomerID]
    , [CompanyName]
    , [ContactName]
    , [Phone]
    , [Fax]
FROM
    [Customers]
WHERE
    (
        [CustomerID] LIKE @sCustomerID
        AND [CompanyName] LIKE @sCompanyName
        AND [ContactName] LIKE @sContactName
    )
ORDER BY
    [CustomerID]
GO

--*****
CREATE PROC [spCustomersInsert]
    @sCustomerID varchar(5)
    ,@sCompanyName varchar(40)
    ,@sContactName varchar(30) = NULL
    ,@sPhone varchar(24) = NULL
    ,@sFax varchar(24) = NULL
AS

```

```

--INSERT record into Customers Table based upon input parameters
--'@sCustomerID', '@sCompanyName', '@sContactName', '@sPhone', '@sFax'
IF NOT EXISTS(SELECT [CustomerID] FROM [Customers] WHERE ([CustomerID] = @sCustomerID))
    BEGIN
        INSERT INTO [Customers]
            (
                [CustomerID]
            , [CompanyName]
            , [ContactName]
            , [Phone]
            , [Fax]
            )
        VALUES
            (
                @sCustomerID
            , @sCompanyName
            , @sContactName
            , @sPhone
            , @sFax
            )
    END
GO

```

```

--*****
CREATE PROC [spCustomersUpdate]
    @sCustomerID varchar(5)
    , @sCompanyName varchar(40)
    , @sContactName varchar(30) = NULL
    , @sPhone varchar(24) = NULL
    , @sFax varchar(24) = NULL
AS
--Update single customer record in Customers Table based upon input parameters
--'@sCustomerID', '@sCompanyName', '@sContactName', '@sPhone', '@sFax'
UPDATE [Customers] SET
    [CompanyName] = @sCompanyName
    , [ContactName] = @sContactName
    , [Phone] = @sPhone
    , [Fax] = @sFax
WHERE
    ([CustomerID] = @sCustomerID)
GO

```

```

--*****
CREATE PROC [spDefectsInsert]
    @sDefectPage varchar(500)

```

```

,@sDefect varchar(1000)
AS
--INSERT a record INTO tblDefects containing Errors occurring within Projects
--web application based upon input parameters '@sDefectPage', '@sDefect'
INSERT INTO [tblDefects]
    ([DefectPage], [Defect])
    VALUES(@sDefectPage, @sDefect)
GO

--*****
CREATE PROC [spCustomersRegionsGroup]
AS
--Group existing regions in the Customers Table
SELECT
    [Region]
FROM
    [Customers]
WHERE [Region] IS NOT NULL
GROUP BY [Region]
ORDER BY [Region]
GO

--*****
CREATE PROC [spCustomersTop10]
    @dtStartDate datetime
    ,@dtEndDate datetime
    ,@sRegion varchar(15)
AS
--SELECT TOP 10 Customers based on (quantity * price) - discount sold
--based on input parameteres, @dtStartDate, @dtEndDate, @sRegion
DECLARE @sSQL varchar(2000)
DECLARE @qt char(1)
DECLARE @eol char(2)
SET @dtEndDate = DateAdd(day, 1, @dtEndDate)
SET @qt = CHAR(39)      --"'"
SET @eol = CHAR(32) + CHAR(10)      --space + new line
SET @sSQL = 'SELECT TOP 10' + @eol
SET @sSQL = @sSQL + '[Customers].[CustomerID], [CompanyName], [Region]' + @eol
SET @sSQL = @sSQL + ', Cast(Sum([UnitPrice]* Cast([Quantity] AS Decimal)' + @eol
SET @sSQL = @sSQL + '* (1.00 - [discount])) AS Decimal(12,2)) AS [SumOrders]' + @eol
SET @sSQL = @sSQL + 'FROM [Customers]' + @eol
SET @sSQL = @sSQL + 'INNER JOIN [Orders] ON' + @eol
SET @sSQL = @sSQL + '[Customers].[CustomerID] = [Orders].[CustomerID]' + @eol
SET @sSQL = @sSQL + 'INNER JOIN [Order Details] ON' + @eol
SET @sSQL = @sSQL + '[Orders].[OrderID] = [Order Details].[OrderID]' + @eol
SET @sSQL = @sSQL + 'WHERE (' + @eol
SET @sSQL = @sSQL + '[OrderDate] BETWEEN ' + @qt + Cast(@dtStartDate AS varchar(20)) + @qt + @eol

```

```
SET @sSQL = @sSQL + 'AND ' + @qt + Cast(@dtEndDate AS varchar(20)) + @qt + @eol
IF (@sRegion != 'ALL' AND Len(@sRegion) > 0)
    SET @sSQL = @sSQL + 'AND [Region] = ' + @qt + @sRegion + @qt + @eol
SET @sSQL = @sSQL + ')' + @eol
SET @sSQL = @sSQL + 'GROUP BY [Customers].[CustomerID], [CompanyName], [Region]' + @eol
SET @sSQL = @sSQL + 'ORDER BY [SumOrders] DESC' + @eol
EXEC (@sSQL)
GO
```