**BHG CODE SNIPPITS**

Declare @tFiscals table(Fiscal int)

Declare @LoopNum as int = 0

DECLARE @tYears table([Year] nvarchar(50))

DECLARE @tMonthNames table([MonthName] nvarchar(50))

DECLARE @tMailBuckets table([MailBucket] nvarchar(50))

WHILE (@LoopNum < 4)

BEGIN

INSERT INTO @tFiscals

SELECT (year(dateadd(m,(@LoopNum \*-1),getdate())) \*100) + month(dateadd(m,(@LoopNum \*-1),getdate()))

SET @LoopNum = @LoopNum +1

END

IF LEN(@Years) > 1

BEGIN

INSERT INTO @tYears([Year])

SELECT item FROM Vision.[wct].[SPLIT](@Years,',')

END

--select \* from @tYears

IF LEN(@MonthNames) > 1

BEGIN

INSERT INTO @tMonthNames([MonthName])

SELECT item FROM Vision.[wct].[SPLIT](@MonthNames,',')

END

--select \* from @tMonthNames

IF LEN(@MailBuckets) > 1

BEGIN

INSERT INTO @tMailBuckets([MailBucket])

SELECT item FROM Vision.[wct].[SPLIT](@MailBuckets,',')

END

UPDATE #Temp SET MinDiff = CAST(DATEDIFF(mi,Date1,Date2) AS DECIMAL(14,4))/1440

DELETE FROM #Temp WHERE MinDiff <= 0

DELETE FROM #Temp WHERE SalesAgent ='Heather McGill'

USE [Vision]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [sub8900].[WorkTime] Script Date: 6/4/2015 1:33:21 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- SELECT [sub8900].[WorkTime]('11/21/2012 07:30:00','11/24/2012 10:30:00')

ALTER FUNCTION [sub8900].[WorkTime]

(

@StartDate DATETIME,

@FinishDate DATETIME

)

RETURNS decimal(14,2)

AS

BEGIN

DECLARE @Temp decimal(14,2)

SET @Temp=0

DECLARE @FirstDay DATE

SET @FirstDay = CONVERT(DATE, @StartDate, 112)

DECLARE @LastDay DATE

SET @LastDay = CONVERT(DATE, @FinishDate, 112)

DECLARE @StartTime TIME

SET @StartTime = CONVERT(TIME, @StartDate)

DECLARE @FinishTime TIME

SET @FinishTime = CONVERT(TIME, @FinishDate)

DECLARE @WorkStart TIME

SET @WorkStart = '08:30'

DECLARE @WorkFinish TIME

SET @WorkFinish = '18:00'

DECLARE @DailyWorkTime decimal(14,2)

SET @DailyWorkTime = DATEDIFF(MINUTE, @WorkStart, @WorkFinish)

IF (@StartTime<@WorkStart)

BEGIN

SET @StartTime = @WorkStart

END

IF (@FinishTime>@WorkFinish)

BEGIN

SET @FinishTime=@WorkFinish

END

DECLARE @CurrentDate DATE

SET @CurrentDate = @FirstDay

DECLARE @LastDate DATE

SET @LastDate = @LastDay

DECLARE @IsHoliday int = 0

WHILE(@CurrentDate<=@LastDate)

BEGIN

SELECT @IsHoliday = COUNT(\*) FROM sub8900.tHoliday WHERE [Date] = @CurrentDate

IF (DATEPART(dw, @CurrentDate) != 1 AND DATEPART(dw, @CurrentDate) != 7 AND @IsHoliday = 0)

BEGIN

IF (@CurrentDate!=@FirstDay) AND (@CurrentDate!=@LastDay)

BEGIN

SET @Temp = @Temp + @DailyWorkTime

END

--IF it starts at startdate and it finishes not this date find diff between work finish and start as minutes

ELSE IF (@CurrentDate=@FirstDay) AND (@CurrentDate!=@LastDay)

BEGIN

SET @Temp = @Temp + DATEDIFF(MINUTE, @StartTime, @WorkFinish)

END

ELSE IF (@CurrentDate!=@FirstDay) AND (@CurrentDate=@LastDay)

BEGIN

SET @Temp = @Temp + DATEDIFF(MINUTE, @WorkStart, @FinishTime)

END

--IF it starts and finishes in the same date

ELSE IF (@CurrentDate=@FirstDay) AND (@CurrentDate=@LastDay)

BEGIN

SET @Temp = DATEDIFF(MINUTE, @StartTime, @FinishTime)

END

END

SET @CurrentDate = DATEADD(day, 1, @CurrentDate)

END

-- Return the result of the function

IF @Temp<0

BEGIN

SET @Temp = 0

END

ELSE

BEGIN

SET @Temp = @Temp / 60.00

END

RETURN @Temp

END

USE [Vision]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [sub8900].[WorkDays] Script Date: 6/4/2015 1:32:20 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- SELECT [sub8900].[WorkDays]('11/01/2012 07:30:00','11/30/2012 10:30:00')

-- SELECT \* FROM sub8900.tHoliday

ALTER FUNCTION [sub8900].[WorkDays]

(

@StartDate DATETIME,

@FinishDate DATETIME

)

RETURNS int

AS

BEGIN

DECLARE @Temp decimal(14,2)

SET @Temp=0

DECLARE @FirstDay DATE

SET @FirstDay = CONVERT(DATE, @StartDate, 112)

DECLARE @LastDay DATE

SET @LastDay = CONVERT(DATE, @FinishDate, 112)

DECLARE @CurrentDate DATE

SET @CurrentDate = @FirstDay

DECLARE @LastDate DATE

SET @LastDate = @LastDay

DECLARE @IsHoliday int = 0

WHILE(@CurrentDate<=@LastDate)

BEGIN

SELECT @IsHoliday = COUNT(\*) FROM sub8900.tHoliday WHERE [Date] = @CurrentDate

IF (DATEPART(dw, @CurrentDate) != 1 AND DATEPART(dw, @CurrentDate) != 7 AND @IsHoliday = 0)

BEGIN

IF (@CurrentDate >= @FirstDay) AND (@CurrentDate <= @LastDay)

BEGIN

SET @Temp = @Temp + 1

END

END

SET @CurrentDate = DATEADD(day, 1, @CurrentDate)

END

-- Return the result of the function

IF @Temp < 0

BEGIN

SET @Temp = 0

END

ELSE

BEGIN

SET @Temp = @Temp

END

RETURN @Temp

END

IF @ReportTyp = 17

BEGIN

INSERT INTO #tTATData

SELECT a8.ApplicationNum,

isnull(dbo.UserName(8900,COALESCE(fa.NewValue,a8.PreClosingQCUserNum),1),' Unassigned') as Rep, --fa.NewValue

COALESCE(lsc.[ItemName],'Undefined') as PrimaryIncomeSource,

fa.ChangeDate AS BeginDate,

ra.ChangeDate AS CompleteDate,

cast(0 as decimal(14,2)) as TotalTime,

a8.CollateralType1,

a8.CollateralType2

FROM sub8900.tApplication8900 a8

Left Join vision.dbo.tApplication a with (NOLOCK) on a.ApplicationNum = a8.ApplicationNum

Inner Join dbo.tGuarantor g on g.SubscriberID = a8.SubscriberID and g.ApplicationNum = a8.ApplicationNum and g.GuarantorNum =1

Inner Join sub8900.tGuarantor8900 g8 on g8.SubscriberID = a8.SubscriberID and g8.ApplicationNum = a8.ApplicationNum and g8.GuarantorNum =1

Left join [tFieldAudit] fa on fa.TableTransientKey = a.TransientKey

and fa.FieldName = 'PreClosingQCUserNum'

and fa.ChangeNum =(Select MIN(y.ChangeNum)

From [tFieldAudit] y

Where y.TableTransientKey = a.TransientKey

and y.FieldName = 'PreClosingQCUserNum')

Left join [tFieldAudit] ra on ra.TableTransientKey = a.TransientKey

and ra.FieldName = 'AssignedToUserNum'

AND ra.OldValue = a8.preclosingqcusernum

and ra.ChangeNum =(Select MIN(x.ChangeNum)

From [tFieldAudit] x

Where x.TableTransientKey = a.TransientKey

and x.FieldName = 'AssignedToUserNum'

AND x.OldValue = a8.preclosingqcusernum)

left join [dbo].[tLookup] lsc on lsc.SubscriberID = 8900 and lsc.ItemData = g8.PrimIncSource and lsc.ItemCategory in ('PrimIncSource')

WHERE fa.ChangeDate >= @StartDate

AND fa.ChangeDate <= @EndDate

END

**HOW to JOIN CTEs**

;with CTETransaction

as

(

  select count(\*)Counts,datepart(year,lt.transactiondate) Years,datepart(Month,lt.transactiondate) Months,Type='Transaction'

  from LeadWarehouse.dbo.LeadTransaction lt

  where lt.transactiondate between '04/01/2013' and '04/01/2015'

  group by datepart(year,lt.transactiondate),datepart(Month,lt.transactiondate)

)

--select \* from ctetransaction c

--order by years,months

,

CTEActivities

as

(

  select count(\*)Counts,datepart(year,act.createddate) Years,datepart(Month,act.createddate) Months,Type='Activity'

  from LeadWarehouse.dbo.activity act

  where act.createddate between '04/01/2013' and '04/01/2015'

  group by datepart(year,act.createddate),datepart(Month,act.createddate)

)

--select \* from CTEActivities c

--order by years,months

,

CTEOpportunity

as

(

  select count(\*)Counts,datepart(year,Opt.createddate) Years,datepart(Month,Opt.createddate) Months,Type='Opportunity'

  from LeadWarehouse.dbo.opportunity Opt

  where Opt.createddate between '04/01/2013' and '04/01/2015'

  group by datepart(year,Opt.createddate),datepart(Month,Opt.createddate)

)

--select \* from CTEOpportunity c

--order by years,months

select \* from CTETransaction ct left join CTEActivities ca on ct.years=ca.years and ct.months=ca.months

left join cteopportunity co on co.years=ct.years and co.months=ca.Months

order by ct.years,ct.Months

**CASE WHEN STMT**

CASE WHEN DateScheduled < @StartDate and a.datecompleted is null THEN 'Prior'

WHEN CAST(COALESCE(a.datecompleted,a.DateScheduled) as Time) < '9:00 AM' THEN '<9am'

WHEN CAST(COALESCE(a.datecompleted,a.DateScheduled) as Time) < '12:00 PM' THEN '9-12pm'

WHEN CAST(COALESCE(a.datecompleted,a.DateScheduled) as Time) < '3:00 PM' THEN '12-3pm'

WHEN CAST(COALESCE(a.datecompleted,a.DateScheduled) as Time) < '6:00 PM' THEN '3pm-6pm'

WHEN CAST(COALESCE(a.datecompleted,a.DateScheduled) as Time) >= '6:00 PM' THEN '>6pm'

ELSE NULL

END AS TimeGroup

**SPLIT FUNCTION CALL**

IF LEN(@ReportType) > 0

BEGIN

INSERT INTO @tDepts(DeptName)

SELECT item FROM Vision.[wct].[SPLIT](@ReportType,',')

END

ALTER function [dbo].[Split](@sText varchar(8000), @sDelim varchar(20) = ' ')

RETURNS @retArray TABLE (idx smallint Primary Key, value varchar(8000))

AS

BEGIN

--modification for BHG to support + and - without space

set @sText = replace(@sText, '+', ' +')

set @sText = replace(@sText, '-', ' -')

set @sText = replace(@sText, '=', '= ')

DECLARE @idx smallint,

@value varchar(8000),

@bcontinue bit,

@iStrike smallint,

@iDelimlength tinyint

IF @sDelim = 'Space'

BEGIN

SET @sDelim = ' '

END

SET @idx = 0

SET @sText = LTrim(RTrim(@sText))

SET @iDelimlength = DATALENGTH(@sDelim)

SET @bcontinue = 1

IF NOT ((@iDelimlength = 0) or (@sDelim = 'Empty'))

BEGIN

WHILE @bcontinue = 1

BEGIN

--If you can find the delimiter in the text, retrieve the first element and

--insert it with its index into the return table.

IF CHARINDEX(@sDelim, @sText)>0

BEGIN

SET @value = SUBSTRING(@sText,1, CHARINDEX(@sDelim,@sText)-1)

BEGIN

INSERT @retArray (idx, value)

VALUES (@idx, @value)

END

--Trim the element and its delimiter from the front of the string.

--Increment the index and loop.

SET @iStrike = DATALENGTH(@value) + @iDelimlength

SET @idx = @idx + 1

SET @sText = LTrim(Right(@sText,DATALENGTH(@sText) - @iStrike))

END

ELSE

BEGIN

--If you can’t find the delimiter in the text, @sText is the last value in

--@retArray.

SET @value = @sText

BEGIN

INSERT @retArray (idx, value)

VALUES (@idx, @value)

END

--Exit the WHILE loop.

SET @bcontinue = 0

END

END

END

ELSE

BEGIN

WHILE @bcontinue=1

BEGIN

--If the delimiter is an empty string, check for remaining text

--instead of a delimiter. Insert the first character into the

--retArray table. Trim the character from the front of the string.

--Increment the index and loop.

IF DATALENGTH(@sText)>1

BEGIN

SET @value = SUBSTRING(@sText,1,1)

BEGIN

INSERT @retArray (idx, value)

VALUES (@idx, @value)

END

SET @idx = @idx+1

SET @sText = SUBSTRING(@sText,2,DATALENGTH(@sText)-1)

END

ELSE

BEGIN

--One character remains.

--Insert the character, and exit the WHILE loop.

INSERT @retArray (idx, value)

VALUES (@idx, @sText)

SET @bcontinue = 0

END

END

END

RETURN

END

**TABLE VARIABLES**

DECLARE @tReps table(RepNum int, RepName nvarchar(50), RepFirstName nvarchar(50))

DECLARE @tLeads table(LeadType nvarchar(50))

INSERT INTO @tReps select 168, 'Lynda Schwartz', 'Lynda', 0

INSERT INTO @tReps select 871, 'Takiyah Campbell', 'Takiyah', 0

INSERT INTO @tReps select 736, 'Russell Christensen', 'Russell', 0

INSERT INTO @tReps select 124, 'Nelson Delpozo', 'Nelson', 0

INSERT INTO @tReps select 223, 'Dachele Dye', 'Dachele', 0

INSERT INTO @tReps select 173, 'Natalie Stewart', 'Natalie', 0

INSERT INTO @tLeads select 'Inbound'

INSERT INTO @tLeads select 'Warm Lead'