**NEW STUFF TO LEARN**

**This code allows us to capture a rolling 12 months of data and converts average minutes to avg hours.**

CREATE PROCEDURE [dbo].[sRptLeadContactTimeRolling12mth] (

       @StartDate as DateTime,

       @EndDate as DateTime

) AS

BEGIN

       SET @StartDate = DATEADD(mm,-11,@EndDate)

       SET @StartDate = DATEADD(dd,-(Day(@StartDate)-1),@StartDate)

       SET @EndDate = DATEADD(SS,-1,DATEADD(DD,1,@EndDate))

       SELECT

              o.OpportunityID,

              o.TransactionID,

              [dbo].[LookupLeadTypeII](o.TransactionID) as Leadtype,

              Year(o.CreatedDate)\*100+Month(o.CreatedDate) as Fiscal,

              Year(o.CreatedDate) as [Year],

              Month(o.CreatedDate) as [Month],

              o.CreatedDate as OpportunityCreateDate,

              a.ActivityID,

              datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted)) as FirstContactAttemptMin,

              cast(datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted))as decimal(14,2)) /cast(60 as decimal(14,2)) as FirstContactAttempt,

              o.AssignedToUserNum

       INTO #YTDLEADS

       FROM Leadwarehouse.dbo.Opportunity o with(nolock)

**This code allows us to capture the beginning of the StartDate day (12am)**

SELECT  EndDate = GETDATE(),

StartDate = DATEADD(dd, - 4, CAST(CONVERT(CHAR(10),GETDATE(),101) as smalldatetime))

**This code allows us to capture 12 months of data from GETDATE() and converts average minutes to avg hours and puts the data into another Table for a job for faster performance.**

DECLARE @BeginFiscal int = year(dateadd(mm,-11,getdate()))\*100 + month(dateadd(mm,-11,getdate()))

       --PRINT(@BeginFiscal)

       INSERT INTO dbo.tOpportunityActivityData

       SELECT o.OpportunityID,

              o.TransactionID,

              cast('' as nvarchar(50)) as LeadType,

              Year(o.CreatedDate)\*100+Month(o.CreatedDate) as Fiscal,

              Year(o.CreatedDate) as [Year],

              Month(o.CreatedDate) as [Month],

              o.CreatedDate as OpportunityCreateDate,

              coalesce(a4.DateCompleted,a3.DateCompleted) as DateCompleted,

              datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted)) as FirstContactAttemptMin,

              a3.ActivityID as ActivityIDa3,

              a4.ActivityID as ActivityIDa4,

              cast(datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted))as decimal(14,2)) /cast(60 as decimal(14,2)) as FirstContactAttempt,

              o.AssignedToUserNum,

              getdate() as RunTime

       --INTO dbo.tOpportunityActivityData

       FROM Leadwarehouse.dbo.Opportunity o with(nolock)

                                                                                                                                         --FIRST ACT with a completed date and created after opp

              Left Outer Join Leadwarehouse.dbo.Activity a3 with(nolock) on a3.activityID =(SELECT TOP 1 activityID FROM Activity with(nolock)

                                                                                                                                                       WHERE ((PID = o.PID and POwner = o.Powner) or TransactionID = o.TransactionID)

                                                                                                                                                       and datecompleted is not null and YEAR(isnull(datecompleted,2000)) >= 1991

                                                                                                                                                       and CreatedDate >= o.CreatedDate

                                                                                                                                                                                  --is not the first act createdby sm or ra

                                                                                                                                                       and ActivityID != ISNULL((SELECT TOP 1 ActivityID FROM Activity with(nolock)

                                                                                                                                                                                                       WHERE TransactionID = o.TransactionID and CreatorUserNum IN (834,765)

                                                                                                                                                                                                       ORDER BY CreatedDate, ActivityID),0)

                                                                                                                                                ORDER BY DateCompleted ASC)

              Left Outer Join Leadwarehouse.dbo.Activity a4 with(nolock) on a4.activityID = o.activityID

                                                                                                                     and a4.CreatorUserNum NOT IN (834,765)

                                                                                                                     and YEAR(isnull(a4.datecompleted,2000)) >= 1991

       WHERE year(o.CreatedDate)\*100 + month(o.CreatedDate) >= @BeginFiscal

       ORDER BY o.OpportunityID

       UPDATE dbo.tOpportunityActivityData SET LeadType = [dbo].[LookupLeadTypeII](TransactionID) WHERE RunTime =(select max(RunTime) from dbo.tOpportunityActivityData)

       DELETE FROM dbo.tOpportunityActivityData WHERE RunTime !=(select max(RunTime) from dbo.tOpportunityActivityData)

       SELECT \* FROM dbo.tOpportunityActivityData

**This code used DATEDIFF and COALESCE and converts minutes into hours.**

datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted)) as FirstContactAttemptMin,

              cast(datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted))as decimal(14,2)) /cast(60 as decimal(14,2)) as FirstContactAttemptHours,

**This code allows us to capture the entire day for the end date when running a report.**

SET @EndDate = dateadd(ss,-1,dateadd(dd,1,@EndDate))

**This code returns a list of the past 6 years to use for an SSRS drop down for Years**

SELECT YEAR(GETDATE()) as [Year]

UNION

SELECT YEAR(GETDATE()) - 1 as [Year]

UNION

SELECT YEAR(GETDATE()) - 2 as [Year]

UNION

SELECT YEAR(GETDATE()) - 3 as [Year]

UNION

SELECT YEAR(GETDATE()) - 4 as [Year]

UNION

SELECT YEAR(GETDATE()) - 5 as [Year]

ORDER BY [Year] DESC

**This code returns a rolling 4 months of data**

Declare @tFiscals table(Fiscal int)

Declare @LoopNum as int = 0

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

THIS CODDE RETREIVES **MONTH TO DATE** DATA

=dateadd("d",-datepart("d",today())+1,today())                              SSRS

=dateadd("d",-datepart("d", getdate())+1,getdate())    TSQL

**SQL Server Rolling 12 Months:**

SELECT sum(TRANSACTION\_AMOUNT)

FROM TRANSACTIONS

WHERE datediff(month, TRANSACTION\_DATE, getdate()) <= 12

AND TRANSACTION\_DATE <= getdate()

SQL Server Month to Date:

SELECT sum(TRANSACTION\_AMOUNT)

FROM TRANSACTIONS

WHERE datediff(month, TRANSACTION\_DATE, getdate()) = 0

AND TRANSACTION\_DATE <= getdate()

SQL Server **Year to Date**:

SELECT sum(TRANSACTION\_AMOUNT)

FROM TRANSACTIONS

WHERE datediff(year, TRANSACTION\_DATE, getdate()) = 0

AND TRANSACTION\_DATE <= getdate()

SQL Server Prior Year:

SELECT sum(TRANSACTION\_AMOUNT)

FROM TRANSACTIONS

WHERE datediff(year, TRANSACTION\_DATE, getdate()) = 1

**How to set up GROUPING in SSRS**

* **USE the CCBalanceTransfers & CCBalanceActivation reports as a template**
* Right click on any field in the detail row.
* Go to “Add Group”, “Row Group”, “Parent Group” under Tablix.
* Group By the first column to be grouped on and choose Add header and Footer.
* Remove the extra old column.
* Delete the bottom record.
* Insert a row above the detail row.
* Fill in the middle row with a darker row.

**NEW SSRS BELOW**

* ="Funded by closer for: " + MonthName(Parameters!Month.Value) + " " + cstr(Parameters!Year.Value)
* This is an expression used for the background color fill
  + =IIF(RunningValue(Fields!CloserName.Value,COUNTDISTINCT,NOTHING) MOD 2,"White","LightGrey")
  + =IIf(RowNumber(Nothing) Mod 2 = 0,"WhiteSmoke","White")
* =DateAdd("d", -6, TODAY())
* =IIF(Parameters!ReportType.Value = 1,"Core Withdrawn Before Submission","Initial Decline")
* = "Days Remaining : " & IIF(DateDiff ("d",Today,"2015-03-10 00:00:00.000")<0,0,DateDiff ("d",Today,"2015-03-10 00:00:00.000"))
* THIS FORMATS AS A PERCENTAGE AND CONCATENATES TWO FIELDS INTO ONE EXPRESSION
* =IIf(Sum(Fields!ResponseCount.Value) = 0, 0, Format(Sum(Fields!ScoredCount.Value) / IIf(Sum(Fields!ResponseCount.Value) = 0, 1, Sum(Fields!ResponseCount.Value)),"P")) & " / " & Sum(Fields!ScoredCount.Value)
* =Month(DateAdd(DateInterval.Minute,-3,Now()))
* =year(today())
* THIS IS USED AS CONDITIONAL FORMATTING FOR A fill EXPRESSION IN A TEXT BOX PROPERTY
  + =Switch(Fields!FundedAmount.Value <= 200000, "Red",Fields!FundedAmount.Value <= 400000, "Orange",Fields!FundedAmount.Value <= 600000, "Yellow",Fields!FundedAmount.Value > 600000, "Green")
* This is used to capture the month to date for the Start Date Parameter
  + =dateadd("d",-datepart("d",today())+1,today())

**DECLARE @tFiscals TABLE (FiscalDesc nvarchar(20), Fiscal int)**

**DECLARE @LoopNum AS int = 0**

**WHILE (@LoopNum < 24)**

**BEGIN**

**INSERT INTO @tFiscals**

**SELECT**

**datename(month, dateadd(m, (@LoopNum \* - 1), getdate())) + ' ' + cast(year(dateadd(m, (@LoopNum \* - 1),**

**getdate())) AS char(4)), (year(dateadd(m,**

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

**SET @LoopNum = @LoopNum + 1 END**

**SELECT \* FROM     @tFiscals ORDER BY Fiscal DESC**

**This code returns a rolling 4 months of data**

Declare @tFiscals table(Fiscal int)

Declare @LoopNum as int = 0

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

**This code allows us to create a larger field when inserting into a #temp table so the value does not get truncated.The first value that goes in might be a smaller value that creates the size.**

cast(NULL as varchar(20)) AS SeniorReviewReason

**This code allows us to capture a rolling 12 months of data and converts average minutes to avg hours.**

CREATE PROCEDURE [dbo].[sRptLeadContactTimeRolling12mth] (

       @StartDate as DateTime,

       @EndDate as DateTime

) AS

BEGIN

       SET @StartDate = DATEADD(mm,-11,@EndDate)

       SET @StartDate = DATEADD(dd,-(Day(@StartDate)-1),@StartDate)

       SET @EndDate = DATEADD(SS,-1,DATEADD(DD,1,@EndDate))

       SELECT

              o.OpportunityID,

              o.TransactionID,

              [dbo].[LookupLeadTypeII](o.TransactionID) as Leadtype,

              Year(o.CreatedDate)\*100+Month(o.CreatedDate) as Fiscal,

              Year(o.CreatedDate) as [Year],

              Month(o.CreatedDate) as [Month],

              o.CreatedDate as OpportunityCreateDate,

              a.ActivityID,

              datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted)) as FirstContactAttemptMin,

              cast(datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted))as decimal(14,2)) /cast(60 as decimal(14,2)) as FirstContactAttempt,

              o.AssignedToUserNum

       INTO #YTDLEADS

       FROM Leadwarehouse.dbo.Opportunity o with(nolock)

**This code allows us to capture the beginning of the StartDate day (12am)**

SELECT  EndDate = GETDATE(),    StartDate = DATEADD(dd, - 4, CAST(CONVERT(CHAR(10),GETDATE(),101) as smalldatetime))

**This code allows us to capture 12 months of data from GETDATE() and converts average minutes to avg hours and puts the data into another Table for a job for faster performance.**

DECLARE @BeginFiscal int = year(dateadd(mm,-11,getdate()))\*100 + month(dateadd(mm,-11,getdate()))

       --PRINT(@BeginFiscal)

       INSERT INTO dbo.tOpportunityActivityData

       SELECT o.OpportunityID,

              o.TransactionID,

              cast('' as nvarchar(50)) as LeadType,

              Year(o.CreatedDate)\*100+Month(o.CreatedDate) as Fiscal,

              Year(o.CreatedDate) as [Year],

              Month(o.CreatedDate) as [Month],

              o.CreatedDate as OpportunityCreateDate,

              coalesce(a4.DateCompleted,a3.DateCompleted) as DateCompleted,

              datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted)) as FirstContactAttemptMin,

              a3.ActivityID as ActivityIDa3,

              a4.ActivityID as ActivityIDa4,

              cast(datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted))as decimal(14,2)) /cast(60 as decimal(14,2)) as FirstContactAttempt,

              o.AssignedToUserNum,

              getdate() as RunTime

       --INTO dbo.tOpportunityActivityData

       FROM Leadwarehouse.dbo.Opportunity o with(nolock)

                                                                                                                                         --FIRST ACT with a completed date and created after opp

              Left Outer Join Leadwarehouse.dbo.Activity a3 with(nolock) on a3.activityID =(SELECT TOP 1 activityID FROM Activity with(nolock)

                                                                                                                                                       WHERE ((PID = o.PID and POwner = o.Powner) or TransactionID = o.TransactionID)

                                                                                                                                                       and datecompleted is not null and YEAR(isnull(datecompleted,2000)) >= 1991

                                                                                                                                                       and CreatedDate >= o.CreatedDate

                                                                                                                                                                                  --is not the first act createdby sm or ra

                                                                                                                                                       and ActivityID != ISNULL((SELECT TOP 1 ActivityID FROM Activity with(nolock)

                                                                                                                                                                                                       WHERE TransactionID = o.TransactionID and CreatorUserNum IN (834,765)

                                                                                                                                                                                                       ORDER BY CreatedDate, ActivityID),0)

                                                                                                                                                ORDER BY DateCompleted ASC)

              Left Outer Join Leadwarehouse.dbo.Activity a4 with(nolock) on a4.activityID = o.activityID

                                                                                                                     and a4.CreatorUserNum NOT IN (834,765)

                                                                                                                     and YEAR(isnull(a4.datecompleted,2000)) >= 1991

       WHERE year(o.CreatedDate)\*100 + month(o.CreatedDate) >= @BeginFiscal

       ORDER BY o.OpportunityID

       UPDATE dbo.tOpportunityActivityData SET LeadType = [dbo].[LookupLeadTypeII](TransactionID) WHERE RunTime =(select max(RunTime) from dbo.tOpportunityActivityData)

       DELETE FROM dbo.tOpportunityActivityData WHERE RunTime !=(select max(RunTime) from dbo.tOpportunityActivityData)

       SELECT \* FROM dbo.tOpportunityActivityData

**This code used DATEDIFF and COALESCE and converts minutes into hours.**

datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted)) as FirstContactAttemptMin,

              cast(datediff(mi,o.CreatedDate,coalesce(a4.DateCompleted,a3.DateCompleted))as decimal(14,2)) /cast(60 as decimal(14,2)) as FirstContactAttemptHours,

**This code allows us to capture the entire day for the end date when running a report.**

SET @EndDate = dateadd(ss,-1,dateadd(dd,1,@EndDate))

**This code deletes unwanted records from a temp table.**

delete from #Leads where FirstContactAttempt is null or FirstContactAttempt < 0 or FirstContactAttempt > 10