DECLARE @FirstDayPreviousMonth DATETIME = DATEADD(mm, DATEDIFF(mm, 0, @AsofDate) - 1, 0)

DECLARE @LastDayPreviousMonth DATETIME = DATEADD(d, - 1, DATEADD(mm, DATEDIFF(mm, 0, @AsofDate), 0))

DECLARE @FirstDayCurrentMonth DATETIME = DATEADD(mm, DATEDIFF(mm, 0, @AsofDate), 0)

DECLARE @LastDayCurrentMonth DATETIME = DATEADD(d, - 1, DATEADD(mm, DATEDIFF(mm, 0, @AsofDate) + 1, 0))

DECLARE @FirstDayPreviousQuarter DATETIME = DATEADD(qq, DATEDIFF(qq, 0, @AsofDate) - 1, 0)

DECLARE @LastDayPreviousQuarter DATETIME = DATEADD(d, - 1, DATEADD(qq, DATEDIFF(qq, 0, @AsofDate), 0))

DECLARE @FirstDayCurrentQuarter DATETIME = DATEADD(qq, DATEDIFF(qq, 0, @AsofDate), 0)

DECLARE @LastDayCurrentQuarter DATETIME = DATEADD(d, - 1, DATEADD(qq, DATEDIFF(qq, 0, @AsofDate) + 1, 0))

CROSS APPLY

select m.member\_id, bcv.Version, 'VERSION NOT MATCH' Reason into #temp\_issues

from members m (nolock)

cross apply ( select benop\_value from

Member\_Benefit\_Events mbe (nolock) where m.member\_id = mbe.member\_id and GETDATE() between mbe.DT\_Effective and mbe.DT\_End

and mbe.DT\_Updated = (select MAX(dt\_updated) from Member\_Benefit\_Events mbe2 where mbe2.Member\_ID = m.Member\_ID and

GETDATE() between DT\_Effective and DT\_End+1)

and mbe.Is\_Verified = 1 and Benop\_Value <> '000' ) be

left join

Member\_Address\_List mal (nolock) on m.Current\_Address\_ID = mal.Address\_ID

PARSE & FORMAT QUARTER

DECLARE @CurrentQuarter INT = DATEPART(QQ, GETDATE())

DECLARE @BenefitStartDate datetime

DECLARE @BenefitEndDate datetime

SET @BenefitStartDate = CASE

WHEN @CurrentQuarter = 1 THEN CAST('1/1/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 2 THEN CAST('4/1/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 3 THEN CAST('7/1/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 4 THEN CAST('10/1/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

END

SET @BenefitEndDate = CASE

WHEN @CurrentQuarter = 1 THEN CAST('3/31/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 2 THEN CAST('6/30/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 3 THEN CAST('9/30/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 4 THEN CAST('12/31/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

END

SUBQUERY on JOIN

select benefit\_event\_id into #benefit\_update

from member\_benefit\_events mbe (nolock)

inner join

( select mbe2.member\_id, MAX(DT\_Updated) as maxdate

from member\_benefit\_events mbe2 (nolock)

left join #members m on mbe2.member\_id = m.member\_id and mbe2.benop\_value = m.Current\_Month\_Benefit

where getdate() between mbe2.dt\_effective and mbe2.dt\_end

group by mbe2.member\_id) u

on mbe.member\_id = u.Member\_ID

and u.maxdate = mbe.dt\_updated

and

mbe.member\_id in

(select member\_id from #members)

SUBQUERY on WHERE clause

join

Benefit\_Plans bp (nolock) on be.Benop\_Value = bp.Plan\_Code and bp.dt\_end > getdate()

join

Catalog\_Export\_Versions bcv (nolock) on be.Benop\_Value = bcv.Plan\_Code

where

m.member\_id in (

select r.member\_ID from [OPTUM\_OTC\_TransactionDB-Hotfix].DBO.[Rec\_Drop6\_Issues] r

where Member\_ID not in ( select [Customer Number] from #tempExclusion))

SUBQUERY on the ON condition

select bp.benefit\_plan, bp.plan\_code as PlanCode, Count(\*) as ActiveMembers

from members m (nolock)

join Member\_Benefit\_Events mbe (nolock) on m.member\_id = mbe.member\_id

and Getdate() between mbe.DT\_Effective and mbe.DT\_End

and mbe.DT\_Updated =

(select MAX(mbe2.dt\_updated) from Member\_Benefit\_Events mbe2 where mbe2.member\_id = m.member\_id

and Getdate() between mbe2.dt\_effective and mbe2.dt\_end+1)

NESTED IF STMTS

IF LEN(@First\_Name) > 1 AND LEN(@Last\_Name) > 0 AND ISNUMERIC(@Phone) = 1 AND LEN(@Phone) = 10 AND ISNUMERIC(@First\_Name) = 0 AND ISNUMERIC(@Last\_Name) = 0 AND LEN(@HIC\_Number) >= 1

BEGIN

SET @RecordResultID=11

END

ELSE

BEGIN

IF LEN(@First\_Name) <= 1 OR LEN(@Last\_Name) <= 0 OR ISNUMERIC(@First\_Name) = 1 OR ISNUMERIC(@Last\_Name) = 1

BEGIN

SET @RecordResultID=13

END

IF ISNUMERIC(@Phone) = 0 OR LEN(@Phone) <> 10 OR @Phone like '%.%'

BEGIN

SET @RecordResultID=15

END

IF LEN(@HIC\_Number) > 12

BEGIN

SET @RecordResultID=17

END

END

Recursive function

CREATE Function [dbo].[Get\_Years](@year datetime)

Returns table

AS

return

declare @year datetime = '20150101'

;With CTE AS

(

Select rowId=1, Cast(Cast(Year(@year)+1 as varchar(4)) + '-01-01' as datetime) DT\_Effective, '2078-12-31 00:00:00.000' DT\_End

union all

Select rowID=rowID+1, DateAdd(year,1,DT\_Effective),DT\_End from CTE where DT\_Effective<=GETDATE()

)

select DT\_Effective,DT\_End, Case When RowId=1 Then @Year else null End DT\_Created

from CTE where DT\_Effective<GETDATE()

GO

Code to loop through a file

SELECT @NextID = MAX(Record\_ID)

FROM Download\_Members\_History (NOLOCK)

WHERE Download\_id=@Download\_ID AND ImportStatus = 'Started'

--Now deal with the rest of the members that were in the file

WHILE (@NextID IS NOT NULL)

BEGIN

SELECT TOP 1 @First\_Name =First\_Name,

@Middle\_Name =Middle\_Name,

@Last\_Name =Last\_Name,

@Member\_ID =Member\_ID,

@Current\_Month\_Benefit =rtrim(Current\_Month\_Benefit),

FROM Download\_Members\_History H (NOLOCK)

SELECT o.order\_id [ORDERNO]

,o.member\_id [CUSTNO]

,o.dt\_order [ENTRYDATE]

,o.dt\_print [EXPORTDATE]

,o.dt\_ship [SHIPDATE]

,CASE WHEN fcm.state IN ('FL', 'MA', 'KS') THEN fcm.state + bp.Source\_Code

ELSE o.Ship\_To\_State + bp.Source\_Code END

as SOURCE\_CODE

,fmn.Flm\_Market\_Name [MarketName]

,RTRIM(m.first\_name) + ' ' + RTRIM(m.last\_name) [NAMEX]

,RTRIM(u.FirstName) [FNAME]

,RTRIM(u.LastName) [LNAME]

,pds.Hcpc\_Code [HCPC\_CODE]

,p.Catalog\_Number [ITEMNO]

,cast(replace(p.Product\_Name,'™','') + ' ' + ISNULL(od.Size\_Code,'') + '' + ISNULL(od.color\_code,'') as varchar(306)) [ITEM\_DESCRIPTION]

,CASE WHEN pds.Shipping\_Conversion\_Item\_No IS NOT NULL THEN pds.Quantity else od.Quantity\_Shipped end SumOfITEM\_QTY

,CASE WHEN pv.Member\_Cost IS NOT NULL THEN od.Quantity\_Shipped \* pv.Member\_Cost ELSE od.Quantity\_Shipped \* p.Total\_Member\_Cost END SumOfEXT\_PRICE

,CASE WHEN pds.Shipping\_Conversion\_Item\_No IS NOT NULL THEN od.quantity\_Shipped \* (pds.Quantity \* pds.Unit\_Cost) else od.Quantity\_Shipped \* pds.unit\_cost end SumOfEXT\_COST

,'U' [BIGSTATUS]

,o.Ship\_to\_state [STATE]

,o.Ship\_to\_state [DEMOGRAPHICS]

,o.Ship\_To\_State [Group#]

,o.Ship\_To\_State [Location]

from

orders o (nolock)

left join

Order\_Details od (nolock) on o.order\_id = od.order\_id

left join

Products p on od.Product\_ID = p.Product\_ID

join

Product\_Dropship\_Item\_Options pds on p.Product\_Code = pds.Product\_Code

and ISNULL(od.Size\_Code,'') = ISNULL(pds.size\_code,'')

and ISNULL(od.Color\_Code,'') = ISNULL(pds.Color\_code,'')

and o.dt\_print between pds.DT\_Effective and pds.DT\_End

left join

dbo.Product\_Dropship\_Member\_Price\_Varianaces pv on pds.Product\_dropship\_item\_option\_id = pv.dropship\_item\_option\_id

left join

Member\_Benefit\_Events mbe (nolock) on o.Benefit\_Event\_ID = mbe.Benefit\_Event\_ID

left join

Benefit\_Plans bp (nolock) on mbe.Benop\_Value = bp.Plan\_Code and o.DT\_Created between bp.DT\_Effective and bp.DT\_End

left join

Members m (nolock) on o.Member\_ID = m.Member\_ID

left join

Flm\_CareManagers fcm (nolock) on fcm.otc\_user\_id = o.Created\_By\_ID

left join

Flm\_Market\_Names fmn (nolock) on fmn.flm\_Source\_Code = CASE WHEN fcm.state IN ('FL', 'MA', 'KS') THEN fcm.state + bp.Source\_Code

ELSE o.Ship\_To\_State + bp.Source\_Code END

left join

svw\_AppSec\_Users u on o.Created\_By\_ID = u.AppSec\_Users\_ID

where

bp.division = 3

and o.dt\_print between ? and ? order by o.Order\_ID

insert into Members\_Group\_Codes

select distinct m.External\_Member\_ID, gc.Group\_code, m.Member\_ID

from Members m

cross apply

(select top 1 group\_code from Download\_Members\_History dmh where dmh.Member\_ID = m.external\_member\_id

order by Download\_ID desc) gc

where m.Member\_ID not in

(select member\_id from Members\_Group\_Codes)

DATE FUNCTIONS

declare @dateString as datetime

declare @CurrentMonthBegin as datetime

declare @CurrentMonthEnd as datetime

declare @NextMonthBegin as datetime

declare @NextMonthEnd as datetime

set @dateString = GETDATE()

--set the benefit date ranges based on the file creation date

select @CurrentMonthBegin=CAST(MONTH(CAST(@dateString as datetime)) as varchar) + '/1/' +

CAST(YEAR(CAST(@dateString as datetime)) as varchar)

select

@CurrentMonthBegin as CurrentMonthBegin,

DateAdd(MM,1,@CurrentMonthBegin)-1 as CurrentMonthEnd,

DateAdd(MM,1,@CurrentMonthBegin) as NextMonthBegin,

DateAdd(MM,2,@CurrentMonthBegin)-1 as NextMonthEnd

SET @BenefitStartDate = CASE

WHEN @CurrentQuarter = 1 THEN CAST('1/1/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 2 THEN CAST('4/1/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 3 THEN CAST('7/1/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 4 THEN CAST('10/1/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

END

SET @BenefitEndDate = CASE

WHEN @CurrentQuarter = 1 THEN CAST('3/31/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 2 THEN CAST('6/30/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 3 THEN CAST('9/30/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

WHEN @CurrentQuarter = 4 THEN CAST('12/31/' + CAST(DATEPART(yyyy, GETDATE()) AS VARCHAR(4)) AS DATETIME)

END

STRING FUNCTIONS

WHERE F.Download\_id=@Download\_ID AND

(

LEN(Member\_ID) < 1 OR

LEN(Mail\_Address\_1) < 1 OR LEN(Mail\_City) < 1 OR LEN(Mail\_State) <> 2 OR LEN(Mail\_Zip) < 5 OR

charindex('0',Mail\_City) > 0 OR charindex('1',Mail\_City) > 0 OR charindex('2',Mail\_City) > 0 OR

charindex('3',Mail\_City) > 0 OR charindex('4',Mail\_City) > 0 OR charindex('5',Mail\_City) > 0 OR

charindex('6',Mail\_City) > 0 OR charindex('7',Mail\_City) > 0 OR charindex('8',Mail\_City) > 0 OR

charindex('9',Mail\_City) > 0 OR charindex('0',Mail\_State) > 0 OR charindex('1',Mail\_State) > 0 OR

charindex('2',Mail\_State) > 0 OR charindex('3',Mail\_State) > 0 OR charindex('4',Mail\_State) > 0 OR

charindex('5',Mail\_State) > 0 OR charindex('6',Mail\_State) > 0 OR charindex('7',Mail\_State) > 0 OR

charindex('8',Mail\_State) > 0 OR charindex('9',Mail\_State) > 0 OR ISNUMERIC(SUBSTRING(Mail\_Zip,1,5)) = 0)

SELECT @Batch\_ID

,SUBSTRING(LineData, 5,9) AS Group\_Number

,SUBSTRING(LineData, 14,9) AS Member\_Number

,RTRIM(SUBSTRING(LineData, 25,12)) AS First\_Name

,SUBSTRING(LineData, 37,1) AS Middle\_Name

,RTRIM(SUBSTRING(LineData, 38,18)) AS Last\_Name

,CASE WHEN ISDATE(SUBSTRING(LineData,167,8)) = 1 THEN CAST(SUBSTRING(LineData, 167,8) AS DATE)

ELSE null -- REPLICATE('0',8)

END AS Date\_Of\_Birth

,SUBSTRING(LineData, 175,1) AS Sex

,SUBSTRING(LineData, 56,30) AS Address1

,SUBSTRING(LineData, 86,30) AS Address2

,SUBSTRING(LineData, 116,15) AS City

,SUBSTRING(LineData, 131,2) AS [State]

,SUBSTRING(LineData, 133,5) AS Zip\_Code

,SUBSTRING(LineData, 146,10) AS Primary\_Phone\_Number

,SUBSTRING(LineData, 176,9) AS SSN

,SUBSTRING(LineData, 193,10) AS HIC\_Number

,SUBSTRING(LineData, 206,12) as [Miscellaneous\_Code\_1]

,SUBSTRING(LineData, 203,1) AS Unknown\_03

,SUBSTRING(LineData, 222,8) AS Effective\_Date

,SUBSTRING(LineData, 230,8) AS Term\_Date

,SUBSTRING(LineData, 242,12) AS [PCP\_ID\_Number]

FROM Import\_Enrollment\_LineData (NOLOCK)

ORDER BY LineID

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FROM

download\_members\_temp dmt

LEFT JOIN

benefit\_plans bp on dmt.current\_month\_benefit = bp.Plan\_code and getdate() between bp.DT\_Effective and (bp.DT\_End + 1)

WHERE

LEN(Member\_ID) < 1 OR

LEN(Mail\_Address\_1) < 1 OR LEN(Mail\_City) < 1 OR LEN(Mail\_State) <> 2 OR LEN(Mail\_Zip) < 5 OR

charindex('0',Mail\_City) > 0 OR charindex('1',Mail\_City) > 0 OR charindex('2',Mail\_City) > 0 OR

charindex('3',Mail\_City) > 0 OR charindex('4',Mail\_City) > 0 OR charindex('5',Mail\_City) > 0 OR

charindex('6',Mail\_City) > 0 OR charindex('7',Mail\_City) > 0 OR charindex('8',Mail\_City) > 0 OR

charindex('9',Mail\_City) > 0 OR charindex('0',Mail\_State) > 0 OR charindex('1',Mail\_State) > 0 OR

charindex('2',Mail\_State) > 0 OR charindex('3',Mail\_State) > 0 OR charindex('4',Mail\_State) > 0 OR

charindex('5',Mail\_State) > 0 OR charindex('6',Mail\_State) > 0 OR charindex('7',Mail\_State) > 0 OR

charindex('8',Mail\_State) > 0 OR charindex('9',Mail\_State) > 0 OR ISNUMERIC(SUBSTRING(Mail\_Zip,1,5)) = 0

OR dmt.current\_month\_benefit IS NULL OR bp.plan\_code is null

UPSERT

IF ISNULL(@Current\_Month\_Benefit,'') <> ''

BEGIN

IF NOT EXISTS (SELECT 1 FROM Member\_Benefit\_Events (NOLOCK) WHERE Member\_ID=@ExistingMember\_ID AND

Benop\_Value=REPLACE(@Current\_Month\_Benefit,'-','') AND

DT\_Effective=@CurrentMonthBegin)

BEGIN --Insert new member benefit

INSERT INTO Member\_Benefit\_Events (Member\_ID, Benop\_Dimension, Benop\_Value, DT\_Effective,

DT\_End, DT\_Created, DT\_Updated, Created\_By\_ID, Updated\_By\_ID, Source\_Process, Update\_Count)

VALUES (@ExistingMember\_ID, 'PBP', REPLACE(@Current\_Month\_Benefit,'-',''),

@CurrentMonthBegin, @CurrentMonthEnd, GetDate(), GetDate(), 1, 1, 'Import', 0)

END

ELSE --Update existing benefit

BEGIN

UPDATE Member\_Benefit\_Events

SET Update\_Count=Update\_Count + 1, DT\_Updated=GetDate(), Updated\_By\_ID=1

WHERE Member\_ID=@ExistingMember\_ID AND

Benop\_Value=REPLACE(@Current\_Month\_Benefit,'-','') AND

DT\_Effective=@CurrentMonthBegin

END

END

SubQuery in join

join Member\_Benefit\_Events mbe (nolock) on m.member\_id = mbe.member\_id

and GETDATE() between mbe.DT\_Effective and mbe.DT\_End

and mbe.DT\_Updated =

(select MAX(dt\_updated) from Member\_Benefit\_Events mbe2 where mbe2.Member\_ID = m.Member\_ID

and GETDATE() between mbe2.DT\_Effective and mbe2.DT\_End+1)

and mbe.Is\_Verified = 1 and Benop\_Value <> '000'