```
-----------------------------------------------------------------------------------------------------------------------------
--CSRC
/*
-- Base Tables
		[serv].[CSRC_TOT_COUNT]  *****
		serv.CSRC_REPORT		 *****
		[serv].[CSRC_AUTO_BASE]  *****
		base.rawZDSVert
		base.placement_details
		base.DATABLOCK_SUMMARY

-- Ancillary Lookup Tables
				serv.CSRC_MODEL_CONVERTER
				[lup].[SDM_HIERARCHY])
				lup.WORK_CENTRE
				serv.csrc_final_areas

-- When is it run?
		-- Wednesday @ 11.30AM
*/

-- Parameters to Change:
print ' Generate Week Date'
declare @WeekDate as date
set @WeekDate = '2019-03-06'
print @WeekDate

print ' Generate Month Date'
declare @BWMonth as date
set @BWMonth = 'MAR 2019'
print @BWMonth

-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
--CSRC SCANS OPPORTUNITES
-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
print 'Total Scans got from CSRC Total File'
SELECT
	[Serial Number]
	,[ERP ID]
		,[Total Counter]
```

```
		,(	[Scanner/FAX:Scan] + [Scanner/FAX:Scan Large])  AS TotalScans
into #TotalCount1
FROM [Management_Information].[serv].[CSRC_TOT_COUNT]


print 'Total Scans by Equioment and serial number'
SELECT
	[Serial Number]
	,[ERP ID]
		,sum([Total Counter]) as [Total Counter]
		,sum(TotalScans) as TotalScans
into #TotalCount2
FROM #TotalCount1
group by
	[Serial Number]
	,[ERP ID]


SELECT
	[Serial Number]
	,[ERP ID]
		,[Total Counter]
		,TotalScans
		,IIF( [Total Counter] <= 0, [Total Counter], TotalScans / [Total Counter]) as ScanningDividedByCounters
into #TotalCount3
FROM #TotalCount2
where IIF( [Total Counter] <= 0, [Total Counter], TotalScans / [Total Counter])  >= 1


select
		row_number() over( partition by Serial_Number order by Serial_Number) as Serial_Count
	,[Equipment]
	,[Serial_Number]
	,[Equipment_Description]
	,[Postal_Code]
	,[Framework_Partner]
	,[Sold_To_Party]
	,[Name_Sold_To]
into #VERTNonDups
from base.RAW_ZDSVERT

--select * from #VERTNonDups
```

```
select * into #UniqueVert
from #VERTNonDups
where Serial_Count = 1

 --select * from #UniqueVert


 drop table serv.FDT_CSRC_Scan_Opps
 SELECT
      [Serial Number]
      ,[ERP ID]
            ,[Total Counter]
            ,TotalScans
            ,ScanningDividedByCounters
            ,z.Equipment
            ,Z.Equipment_Description
            ,Z.Framework_Partner
            ,z.Name_Sold_To
            ,z.Sold_To_Party
            ,z.Postal_Code
into serv.FDT_CSRC_Scan_Opps
FROM #TotalCount3 as k
left join #UniqueVert as z on k.[Serial Number] = z.[Serial_Number]
WHERE z.Equipment is not null



-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--CCSRC WEEKLY FULL DATA
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-- Add any new models to the CSRC_MODEL_CONVERTER

select distinct
            Product
into #MissingModels
from serv.CSRC_REPORT
where Product
```

```
	not in(select Model from serv.CSRC_MODEL_CONVERTER)
--select * from #MissingModels


-- Declare the cursor
declare @Product nvarchar(50)
declare @Autonum int
declare @counter as int
select @Autonum = max([auton]) from serv.CSRC_MODEL_CONVERTER
declare missing_cursor cursor for
select Product from #MissingModels

-- Open the cursor
	open missing_cursor
	----
	fetch next from missing_cursor
			into @Product
	---- Do something >>>
	set @counter = 1
	while @@FETCH_STATUS = 0
			begin
				insert into
					serv.CSRC_MODEL_CONVERTER ([Model], [Model Type],[auton])
					values(@Product,
						case
							when @Product like '%PRO%'
	or @Product like '%PRESS%'

	or @Product like '%Accurio%'
							then 'PPD'
							else 'MFD'
						end
					,@Autonum + @counter)
					set @counter = @counter + 1
					fetch next from missing_cursor into @Product
			end
	----
	close missing_cursor
	deallocate missing_cursor
	go
```

```sql
-- Check for model dups
select
	*
into #modelDupCount
from
(
	select
		model
		,count(model) as mc
	from serv.CSRC_MODEL_CONVERTER
	group by model
	having count(model) >1
) as tc

-- sort out dups if necessary or move on

declare @dupcount int
select @dupcount = count(*) from #modelDupCount
print @dupcount

if @dupcount = 0
				goto Proceed
else
				select
					model
					, [Model Type]
					, auton
					, iif(row_number() over(partition by model order by model)>1,0,1) as intnum
				into #dupRemover
				from serv.CSRC_MODEL_CONVERTER
				--select * from #dupRemover

				select
					model
					, [Model Type]
					, auton
				into #dupRemover2
				from #dupRemover
				where intnum = 1
				--select * from #dupRemover2
```

```
                        delete from serv.CSRC_MODEL_CONVERTER

                        select model, [Model Type], auton
                        into serv.CSRC_MODEL_CONVERTER
                        from #dupRemover2


Proceed:


-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-- Generate CSRC Final datasets

-- ---------------------Sort out auto so no dups

select
[Pool]
      ,[Customer]
      ,[Information]
      ,[Postal Code]
      ,[Communication type]
      ,[Create restock order automatically]
      ,[Equipment]
      ,[Machine type]
      ,[Contact person for pool]
      ,[Telephone Number]
      ,[Send email confirmation]
      ,[Planning plant]
      ,[Planner group]
      ,[Sales Organization]
      ,[Distribution Channel]
      ,[no AutoRestock due to customer request]
            ,iif(row_number()over(partition by Equipment order by Equipment) > 1,0,1) as UniqueID
into #AutoDeDupe1
FROM [Management_Information].[serv].[CSRC_AUTO_BASE]

select
[Pool]
      ,[Customer]
      ,[Information]
      ,[Postal Code]
      ,[Communication type]
```

```
      ,[Create restock order automatically]
      ,[Equipment]
      ,[Machine type]
      ,[Contact person for pool]
      ,[Telephone Number]
      ,[Send email confirmation]
      ,[Planning plant]
      ,[Planner group]
      ,[Sales Organization]
      ,[Distribution Channel]
      ,[no AutoRestock due to customer request]
into #AutoWorkingData
FROM #AutoDeDupe1
where UniqueID = 1


-- ----------------------Create list of OPS devices

select
            Equipment
            ,[Name_Sold_to]
            ,[Serial_Number]
            ,[Cust_Hierarchy]
into #OPS_Devices
from base.RAW_ZDSVERT
where [Cust_Hierarchy] in (select distinct [HEIRARCHY] from [lup].[SDM_HIERARCHY])


-- ----------------------Create list of gma devices

select
            Equipment
            ,[Name_Sold_to]
            ,[Serial_Number]
            ,[Cust_Hierarchy]
into #GMA_Devices
from base.RAW_ZDSVERT
where [Beu_ID] is not null
-- select * from #GMA_Devices

-- ----------------------Create list of volume per month from placement details
```

```
select
		Equipment
		,cast([Click_BW_Avg] as int) as Avg_Click_BW_Per_Month
		,cast([Click_Color_Avg] as int) as Avg_Click_Colour_Per_Month
		,cast([Click_BW_Avg] as int) + cast([Click_Color_Avg] as int)  as Avg_Click_Total_Per_Month
		,w.[Work_Centre]
		,w.[Serv_Reg_Desc]
		,w.[Serv_Dist_Desc]
into #CSRC_PLacement_Details
from [base].[PLACEMENT_DETAILS] as p
left join lup.WORK_CENTRE as w on IIF(len(p.[Work_Centre_Code])>5,substring(p.[Work_Centre_Code],6,5),p.[Work_Centre_Code]) = w.[Work_Centre]
--drop table #Monthly_Volumes_CSRC


-- ----------------------Create list of vert fields needed

select
		Equipment
		,[Serial_Number]
		,[Framework_Partner]
		,[Postal_Code]
		,Ship_to_Party
into #CSRC_VERT
from base.RAW_ZDSVERT
-- drop table #CSRC_VERT


-- ----------------------Create weekly CSRC check file

SELECT
			[Device Name]
			,p.Avg_Click_Total_Per_Month
			,p.Avg_Click_Colour_Per_Month
			,p.Avg_Click_BW_Per_Month
			,p.Serv_Reg_Desc
			,f.[Final Service District]
			,p.Work_Centre
			,v.Framework_Partner
			,v.Postal_Code
			,v.Ship_to_Party
			,[Center ID]
			,[Com Server]
```

```
                ,[Device ID]
                ,[Serial Number]
                ,[Product] as ModelDescription
                ,m.[Model Type]
                ,[Location]
                ,[IP Address]
                ,[MAC Address]
                ,[Comm Type]
                ,a.[Customer]
                ,[Service Office]
                ,[Sales Office]
                ,f.[SupportedBy]
                ,f.[MFDPPD]
                ,[Inspection Settings]
                ,[Registing]
                ,[Mount Date]
                ,[Reg Date]
                ,[Init Date]
                ,[Latest Received]
                ,[Latest Counter Date]
                ,[PeriodicJob]
                ,[STATUS]
                ,
                        case
                                when k.[Create restock order automatically] is null then 'Flag is Not Set'
                                else k.[Create restock order automatically]
                        end as AutoRestocking
                ,
                        case
                                when k.[no AutoRestock due to customer request] is null then ''
                                else k.[no AutoRestock due to customer request]
                        end as [Customer Request no AutoRestock]

                ,[Customer Email]
                ,[Service Office Email]
                ,[AutoNotify]
                ,[Controler]
                ,[Under Maintenance]
into #CSRC_Weekly_1
  FROM [Management_Information].[serv].[CSRC_REPORT] as a
  left join serv.CSRC_MODEL_CONVERTER as m on a.[Product] = m.Model
  left join #CSRC_PLacement_Details as p on a.[Device Name] = p.Equipment
```

```
  left join #CSRC_VERT as v on a.[Device Name] = v.Equipment
  left join #AutoWorkingData as k on a.[Device Name]= k.Equipment
  left join [serv].[CSRC_FINAL_AREAS] as f on p.Work_Centre = f.[Work_Centre]
  -- select * from #CSRC_Weekly_1
  -- select * from #CSRC_PLacement_Details

drop table serv.CSRC_Weekly_Final
   SELECT
                        --@WeekDate as CurrentWeek
                        [Device Name]
                        ,Avg_Click_Total_Per_Month
                        ,Avg_Click_Colour_Per_Month
                        ,Avg_Click_BW_Per_Month
                        ,Serv_Reg_Desc
                        ,[Final Service District]
                        ,Work_Centre
                        ,Framework_Partner
                        ,Postal_Code
                        ,Ship_to_Party
                        ,[Center ID]
                        ,[Com Server]
                        ,[Device ID]
                        ,[Serial Number]
                        ,ModelDescription
                        ,[Model Type]
                        ,[Location]
                        ,[IP Address]
                        ,[MAC Address]
                        ,[Comm Type]
                        ,[Customer]
                        ,[Service Office]
                        ,[Sales Office]
                        ,[SupportedBy]
                        ,[MFDPPD]
                        ,[Inspection Settings]
                        ,[Registing]
                        ,[Mount Date]
                        ,[Reg Date]
                        ,[Init Date]
                        ,[Latest Received]
                        ,[Latest Counter Date]
                        ,[PeriodicJob]
```

```
		,[STATUS]
		,AutoRestocking
		,[Customer Request no AutoRestock]
		,
			case
				when AutoRestocking = 'Not set up' then 'Not set up'
				when AutoRestocking = 'Flag is not set' and [Customer Request no AutoRestock] is null then 'Set up - Not Actioned'
				when AutoRestocking = 'Flag is not set' and [Customer Request no AutoRestock] = 'X' then 'Requested to have no auto-restock'
				else 'Set up on auto-restock'
			end as AutoRestocking1

		,[Customer Email]
		,[Service Office Email]
		,[AutoNotify]
		,[Controler]
		,[Under Maintenance]
into serv.CSRC_Weekly_Final
FROM #CSRC_Weekly_1
-- select * from  serv.CSRC_Weekly_Final
-- select distinct AutoRestocking from #CSRC_Weekly_1

update serv.CSRC_Weekly_Final
set Avg_Click_Colour_Per_Month = 1000 where Avg_Click_Colour_Per_Month is null
update serv.CSRC_Weekly_Final
set Avg_Click_BW_Per_Month = 1000 where Avg_Click_BW_Per_Month is null
update serv.CSRC_Weekly_Final
set Avg_Click_Total_Per_Month = 2000 where Avg_Click_Total_Per_Month is null



-------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-- Generate CSRC checking files using CSRC Weekly


-- ----------------------Create OFFLINE CCS

drop table serv.CSRC_Offline_CCS_Devices
select
	*
into serv.CSRC_Offline_CCS_Devices
from serv.CSRC_Weekly_Final
where
```

```
		Framework_Partner in( 'FMP-GPS' , 'FMP-GPS3' ,'FMP-RM3781' ,'FMP-3781L3')
		and [STATUS] in('OFFLINE','INI')
order by Avg_Click_Colour_Per_Month desc
-- drop table serv.CSRC_Offline_CCS_Devices


-- ----------------------Create offline dca

drop table serv.CSRC_Offline_DCA_Devices
select
		*
into serv.CSRC_Offline_DCA_Devices
from serv.CSRC_Weekly_Final
where
		[Comm Type] LIKE '%DCA%' and ([STATUS] = 'OFFLINE' or [STATUS] = 'INI')
order by Avg_Click_Colour_Per_Month desc


-- ----------------------Create offline but other device onsite online


select
		Customer
		,Location
		,[STATUS]
into #All_Customers_Online_INI
from serv.CSRC_Weekly_Final
where [STATUS] in('INI','ONLINE')



drop table serv.CSRC_Offline_OnlineOtherSite_Devices
select
		*
into serv.CSRC_Offline_OnlineOtherSite_Devices
from serv.CSRC_Weekly_Final
where
		[Comm Type] LIKE '%DCA%' and ([STATUS] = 'OFFLINE' or [STATUS] = 'INI')
order by Avg_Click_Colour_Per_Month desc

-- select * from serv.CSRC_Offline_OnlineOtherSite_Devices
```

```sql
-- ----------------------Create offline PP Devices


drop table serv.CSRC_Offline_PPD_Devices
select
	*
into serv.CSRC_Offline_PPD_Devices
from serv.CSRC_Weekly_Final
where
	[Model Type] like '%PPD%' and ([STATUS] = 'OFFLINE' or [STATUS] = 'INI')
order by Avg_Click_Colour_Per_Month desc


-- ----------------------Create items on CSRC but not VERT
drop table serv.CSRC_NOT_ON_VERT
select
	*
into serv.CSRC_NOT_ON_VERT
from serv.CSRC_Weekly_Final
where
	[Device Name] not in ( select distinct Equipment from base.RAW_ZDSVERT)
order by Avg_Click_Colour_Per_Month desc


-- ----------------------Create last counter date
select
	[Device Name]
	,Customer
	,[Serial Number]
	,[Model Type]
	,[Comm Type]
	,[STATUS]
	,Avg_Click_Colour_Per_Month
	,cast([Latest Counter Date] as datetime) as [Latest Counter Date]
into #CSRC_Last_Counter_Date_1
from serv.CSRC_Weekly_Final
order by Avg_Click_Colour_Per_Month desc
-- select * from serv.CSRC_Last_Counter_Date


drop table serv.CSRC_Last_Counter_Date
```

```sql
select
	[Device Name]
	,Customer
	,[Serial Number]
	,[Model Type]
	,[Comm Type]
	,[STATUS]
	,Avg_Click_Colour_Per_Month
	,[Latest Counter Date]
	,
	case
		when [Latest Counter Date] = '1900-01-01 00:00:00.000' then 'No Latest Counter Date Found'
		when getdate() - [Latest Counter Date] < 7 then 'Within Last Week'
		when getdate() - [Latest Counter Date] < 14 then 'Within Last Two Week'
		when getdate() - [Latest Counter Date] < 30 then 'Within Last Month'
		when getdate() - [Latest Counter Date] < 90 then 'Within Last 3 Months'
		when getdate() - [Latest Counter Date] < 180 then 'Within Last 6 Months'
		when getdate() - [Latest Counter Date] < 270 then 'Within Last 9 Months'
		when getdate() - [Latest Counter Date] < 360 then 'Within Last Year'
		when getdate() - [Latest Counter Date] < 720 then 'Within Last 2 Years'
		else '> 2 Years'
	end
	as Last_Date_Banding

into serv.CSRC_Last_Counter_Date
from #CSRC_Last_Counter_Date_1
order by Avg_Click_Colour_Per_Month desc

-- select * from serv.CSRC_Last_Counter_Date



-------------------------------------------------------------------------------------------------------------------------------------------------------------
-- Generate CSRC historic data


declare @WeekDate2 as date
set @WeekDate2 = '2019-01-28'
declare @BWMonth2 as date
set @BWMonth2 = 'DEC 2018'

select
```

```
            @WeekDate2 as CurrentWeek
           ,[Equipment]
into #Historic_MIF_1
from base.DATABLOCK_SUMMARY
where Month_Year = @BWMonth2


-- ----------------------Link to get whether the equipmnet is gma, ops or csrc
select
            CurrentWeek
           ,a.[Equipment]
           ,o.[Cust_Hierarchy] as OPS_Device
           ,c.[STATUS] as CSRC_Device
           ,g.[Cust_Hierarchy] as GMA_Device
into #Historic_MIF_2
from #Historic_MIF_1 as a
left join #OPS_Devices as o on a.Equipment = o.Equipment
left join serv.CSRC_Weekly_Final as c on a.Equipment = c.[Device Name]
left join #GMA_Devices as g on a.Equipment = g.Equipment
-- select * from #Historic_MIF_2
-- drop table #Historic_MIF_2


------------------------turn to 1's and 0's
select
            CurrentWeek
           ,a.[Equipment]
           ,iif(OPS_Device is not null, 1,0) as OPS_Device
           ,iif(CSRC_Device is not null, 1, 0) as CSRC_Device
           ,iif(GMA_Device is not null, 1, 0) as GMA_Device
into #Historic_MIF_3
from #Historic_MIF_2 as a
left join #OPS_Devices as o on a.Equipment = o.Equipment
left join serv.CSRC_Weekly_Final as c on a.Equipment = c.[Device Name]
left join #GMA_Devices as g on a.Equipment = g.Equipment
-- Drop table #Historic_MIF_3
-- select * from #Historic_MIF_3


------------------------Add in mif not on anything
select
            CurrentWeek
```

```
		,[Equipment]
		,OPS_Device
		,CSRC_Device
		,GMA_Device
		,iif(OPS_Device = 0 and CSRC_Device = 0 and GMA_Device = 0, 1, 0) as Not_On_Anything
into #Historic_MIF_4
from #Historic_MIF_3 as a
-- select * from #Historic_MIF_4
-- drop table #Historic_MIF_4


------------------------Pivot up the status for all CSRC devices
select
		*
into #Status_Pivot
from
		(select * from
		(select
			[Device Name]
			,[STATUS]
			,cast(1 as int) as rowCounts
		from serv.CSRC_Weekly_Final as p) as BaseData
		pivot(
			sum(rowCounts)
			for [STATUS] in([ONLINE],[OFFLINE],[INI])
		) as pvt) as bd
-- select * from #Status_Pivot
-- drop table #Status_Pivot


------------------------Pivot up the status for all Lisa's team devices
select
		*
into #Lisa_Team_Pivot
from
		(select * from
		(select
			[Device Name]
			,AutoRestocking1
			,1 as rowCounts
		from serv.CSRC_Weekly_Final as p) as BaseData
		pivot(
```

```
                    sum(rowCounts)
                    for AutoRestocking1 in([Requested to have no auto-restock],[Set up on auto-restock],[Set up - Not Actioned])
          ) as pvt) as bd
-- select * from #Lisa_Team_Pivot
-- drop table #Lisa_Team_Pivot


------------------------link two pivots back to historic dataset & add on work cetnres and service districts
select
          CurrentWeek
          ,b.[Final Service District] as Service_District
          ,a.[Equipment]
          ,iif(OPS_Device = 1 and CSRC_Device = 1,0,OPS_Device) as OPS_Device
          ,CSRC_Device
          ,IIF(GMA_Device = 1 and CSRC_Device = 1,0,GMA_Device) as GMA_Device
          ,Not_On_Anything
          ,IIF(S.INI is null, 0, 1) as [INI]
          ,IIF(S.[OFFLINE] is null, 0, 1) as [OFFLINE]
          ,IIF(S.[ONLINE] is null, 0, 1) as [ONLINE]
          ,iif(l.[Requested to have no auto-restock] is null,0,1) as [Requested to have no auto-restock]
          ,iif(l.[Set up - Not Actioned] is null,0,1) as [Set up - Not Actioned]
          ,iif(l.[Set up on auto-restock] is null, 0, 1) as [Set up on auto-restock]
into #Historic_MIF_5
from #Historic_MIF_4 as a
left join #Status_Pivot as s on a.Equipment = s.[Device Name]
left join #Lisa_Team_Pivot as l on a.Equipment = l.[Device Name]
left join #CSRC_PLacement_Details as p on a.Equipment = p.Equipment
left join [serv].[CSRC_FINAL_AREAS] as b on p.Work_Centre = b.Work_Centre

-- select * from #Historic_MIF_5
-- drop table #Historic_MIF_5
--select [Equipment] from  #Historic_MIF_5 where Equipment not in(select distinct [Device Name] from #Status_Pivot)

--select [Device Name] from #Status_Pivot
--where
--          [Device Name] in(select distinct Equipment from #Historic_MIF_5)
--and
--          CSRC_Device = 1


------------------------link two pivots back to historic dataset & add on work cetnres and service districts
select
```

```
		CurrentWeek
		,Service_District
		,sum(OPS_Device) + sum(CSRC_Device) + sum(GMA_Device) + sum(Not_On_Anything) as TotalMIF
		,sum(OPS_Device) + sum(GMA_Device) + sum(Not_On_Anything) as NotonCSRCMIF
		,sum(OPS_Device) as OPS_Device
		,sum(CSRC_Device) as CSRC_Device
		,sum(GMA_Device) as GMA_Device
		,sum(Not_On_Anything) as Not_On_Anything
		,sum([INI]) as CSRC_INI
		,sum([OFFLINE]) as CSRC_Offline
		,sum([ONLINE]) as CSRC_Online
		,sum([Requested to have no auto-restock]) as [Requested to have no auto-restock]
		,sum([Set up - Not Actioned]) as [Set up - Not Actioned]
		,sum([Set up on auto-restock]) as [Set up on auto-restock]
into #Historic_MIF_6
from #Historic_MIF_5 as a
where Service_District is not null or Service_District not like '%Dealer%'
group by
		CurrentWeek
		,Service_District
-- select * from #Historic_MIF_6
-- drop table #Historic_MIF_6


------------------------create pre dataset for lookup to Lisa's team
select
		[Final Service District]
	,[SupportedBy]
	,[MFDPPD]
into #Lisa_Lookup_List
from [serv].[CSRC_FINAL_AREAS]
group by		[Final Service District]
	,[SupportedBy]
	,[MFDPPD]


------------------------lookup for lookup to Lisa's team
select
		CurrentWeek
		,Service_District
		,l.[SupportedBy]
		,l.[MFDPPD]
```

```
		,Not_On_Anything
		,NotonCSRCMIF
		,TotalMIF
		,CSRC_Device
		,OPS_Device
		,GMA_Device
		,CSRC_INI
		,CSRC_Offline
		,CSRC_Online
		,[Requested to have no auto-restock]
		,[Set up - Not Actioned]
		,[Set up on auto-restock]
into #Historic_MIF_Final
from #Historic_MIF_6 as a
left join #Lisa_Lookup_List as l on a.Service_District = l.[Final Service District]
-- select * from #Historic_MIF_Final
-- drop table #Historic_MIF_Final

select sum([ONLINE]) from #Status_Pivot
select sum([OFFLINE]) from #Status_Pivot
select sum([INI]) from #Status_Pivot
select * from #Historic_MIF_Final


------------------------Add the weeks data into the historic table

declare @WeeksData date
select @WeeksData = max(CurrentWeek) from #Historic_MIF_Final

declare @CurrentWeekCheck int
select @CurrentWeekCheck = count(*) from (select distinct CurrentWeek from #Historic_MIF_Final where Max(CurrentWeek) in(select distinct Date_Week from
[serv].[CSRC_HISTORIC_DATA])) as a

if @CurrentWeekCheck > 0
	begin
		print 'The week already exists in the historic table'
		goto SkipSection
	end
else
	begin
			insert into  [serv].[CSRC_HISTORIC_DATA]
			select * from #Historic_MIF_Final
```

```
        end


SkipSection:
```