**SET** ThousandSep=' ';

**SET** DecimalSep='.';

**SET** MoneyThousandSep=' ';

**SET** MoneyDecimalSep=',';

**SET** MoneyFormat='R # ##0,00;R-# ##0,00';

**SET** TimeFormat='hh:mm:ss TT';

**SET** DateFormat='YYYY/MM/DD';

**SET** TimestampFormat='YYYY/MM/DD hh:mm:ss[.fff] TT';

**SET** MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';

**SET** DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';

**Section** Access;

*[Security Users]:*

**LOAD** \* **INLINE** [

;

**Concatenate**

Directory;

**LOAD**

[ACCESS],

[USERID],

[PASSWORD],

[SERIAL],

[NTNAME],

[ACTIVEUSER]

**FROM** Users.xls (biff, embedded labels, table is [Sheet1$]);

**Section** Application;

**inputfield** Reconciled;

**CONNECT** **TO** [Provider=SQLOLEDB.1;Persist Security Info=True;User ID=sa;Initial Catalog=ibolt31;Data Source=recibolt2;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=RECQVIEW;Use Encryption for Data=False;Tag with column collation when possible=False;MARS Connection=False;DataTypeCompatibility=0;Trust Server Certificate=False] (XPassword is xxx);

*[Details]:*

**Load**

EntryNumber,

FileName,

EventDate,

EventType,

'No' **as** Reconciled,

DayStart(EventDate) **as** EventDay,

Upper(BranchCode) **as** BranchCode,

MonthName(EventDate) **as** MonthName,

Day(EventDate) **as** DayName,

ErrorMsg,

if(subfield(FileName, '\_' ,2)='transaction',if(isnum(mid(ErrorMsg,index( ErrorMsg, ':' )+1,9999)), mid(ErrorMsg,index( ErrorMsg, ':' )+1,9999),''),'') **as** iBoltSAPKey,

if(subfield(FileName, '\_' ,2)='transaction' **and** subfield(FileName, '\_' ,3)='1','GRV',if(subfield(FileName, '\_' ,2)='transaction' **and** subfield(FileName, '\_' ,3)='2','DEL',if(subfield(FileName, '\_' ,2)='transaction' **and** subfield(FileName, '\_' ,3)='9','ROS',if(subfield(FileName, '\_' ,2)='transaction' **and** subfield(FileName, '\_' ,3)='10' **and** mid(subfield(FileName, '\_' ,4),7,2)='02','ROS', if(subfield(FileName, '\_' ,2)='transaction' **and** subfield(FileName, '\_' ,3)='10' **and** mid(subfield(FileName, '\_' ,4),7,2)='01','ROP'))))) **as** iBoltSourceDocType,

XMLData,

SuccessIndicator **as** xxGood,

Direction **as** xxDirection,

Upper(subfield(FileName, '\_' ,1)) **as** FN\_Branch\_Div\_Code,

subfield(FileName, '\_' ,2) **as** FN\_Transaction\_Type,

if(substringcount(FileName,'\_')=2, mid(subfield(FileName, '\_' ,3),1,4) & '-' & mid(subfield(FileName, '\_' ,3),5,2) & '-' & mid(subfield(FileName, '\_' ,3),7,2) & ' ' & mid(subfield(FileName, '\_' ,3),9,2) & ':' & mid(subfield(FileName, '\_' ,3),11,2) & ':' & mid(subfield(FileName, '\_' ,3),13,2), if(substringcount(FileName,'\_')=3, mid(subfield(FileName, '\_' ,4),1,4) & '-' & mid(subfield(FileName, '\_' ,4),5,2) & '-' & mid(subfield(FileName, '\_' ,4),7,2) & ' ' & mid(subfield(FileName, '\_' ,4),9,2) & ':' & mid(subfield(FileName, '\_' ,4),11,2) & ':' & mid(subfield(FileName, '\_' ,4),13,2),if(substringcount(FileName,'\_')=4,mid(subfield(FileName, '\_' ,5),1,4) & '-' & mid(subfield(FileName, '\_' ,5),5,2) & '-' & mid(subfield(FileName, '\_' ,5),7,2) & ' ' & mid(subfield(FileName, '\_' ,5),9,2) & ':' & mid(subfield(FileName, '\_' ,5),11,2) & ':' & mid(subfield(FileName, '\_' ,5),13,2),'Error'))) **as** FileDate,

if(substringcount(FileName,'\_')=2, mid(subfield(FileName, '\_' ,3),1,4) & '-' & mid(subfield(FileName, '\_' ,3),5,2) & '-' & mid(subfield(FileName, '\_' ,3),7,2) & ' ' & mid(subfield(FileName, '\_' ,3),9,2) & ':' & mid(subfield(FileName, '\_' ,3),11,2) & ':' & mid(subfield(FileName, '\_' ,3),13,2), if(substringcount(FileName,'\_')=3, mid(subfield(FileName, '\_' ,4),1,4) & '-' & mid(subfield(FileName, '\_' ,4),5,2) & '-' & mid(subfield(FileName, '\_' ,4),7,2) & ' ' & mid(subfield(FileName, '\_' ,4),9,2) & ':' & mid(subfield(FileName, '\_' ,4),11,2) & ':' & mid(subfield(FileName, '\_' ,4),13,2),if(substringcount(FileName,'\_')=4,mid(subfield(FileName, '\_' ,5),1,4) & '-' & mid(subfield(FileName, '\_' ,5),5,2) & '-' & mid(subfield(FileName, '\_' ,5),7,2) & ' ' & mid(subfield(FileName, '\_' ,5),9,2) & ':' & mid(subfield(FileName, '\_' ,5),11,2) & ':' & mid(subfield(FileName, '\_' ,5),13,2),'Error'))) **as** [Doc Date],

if(subfield(FileName, '\_' ,2)='account' **or** subfield(FileName, '\_' ,2)='accountconfirmation' **or** subfield(FileName, '\_' ,2)='product', Left(subfield(FileName, '\_' ,4),6),'') **as** FileTime,

if(subfield(FileName, '\_' ,2)='transactioncasualsummary' **or** subfield(FileName, '\_' ,2)='paymentcasualsummary', subfield(FileName, '\_' ,3),'') **as** FN\_BatchNo,

if(subfield(FileName, '\_' ,2)='cashrequisition' **or** subfield(FileName, '\_' ,2)='cashreceipt', subfield(FileName, '\_' ,3),'') **as** FN\_TransCode,

if(subfield(FileName, '\_' ,2)='transaction', subfield(FileName, '\_' ,3),'') **as** FN\_Trans\_Code,

if(Right(if(subfield(FileName, '\_' ,2)='transaction', subfield(FileName, '\_' ,4),''),1)='R',mid(if(subfield(FileName, '\_' ,2)='transaction', subfield(FileName, '\_' ,4),''),1,len(if(subfield(FileName, '\_' ,2)='transaction', subfield(FileName, '\_' ,4),''))-1),if(subfield(FileName, '\_' ,2)='transaction', subfield(FileName, '\_' ,4),'')) **as** YMSTicketNo,

if(subfield(FileName, '\_' ,2)='hastotal', subfield(FileName, '\_' ,3),'') **as** xxLastUpdate,

if(Now()-(if(substringcount(FileName,'\_')=2, mid(subfield(FileName, '\_' ,3),1,4) & '-' & mid(subfield(FileName, '\_' ,3),5,2) & '-' & mid(subfield(FileName, '\_' ,3),7,2) & ' ' & mid(subfield(FileName, '\_' ,3),9,2) & ':' & mid(subfield(FileName, '\_' ,3),11,2) & ':' & mid(subfield(FileName, '\_' ,3),13,2), if(substringcount(FileName,'\_')=3, mid(subfield(FileName, '\_' ,4),1,4) & '-' & mid(subfield(FileName, '\_' ,4),5,2) & '-' & mid(subfield(FileName, '\_' ,4),7,2) & ' ' & mid(subfield(FileName, '\_' ,4),9,2) & ':' & mid(subfield(FileName, '\_' ,4),11,2) & ':' & mid(subfield(FileName, '\_' ,4),13,2),if(substringcount(FileName,'\_')=4,mid(subfield(FileName, '\_' ,5),1,4) & '-' & mid(subfield(FileName, '\_' ,5),5,2) & '-' & mid(subfield(FileName, '\_' ,5),7,2) & ' ' & mid(subfield(FileName, '\_' ,5),9,2) & ':' & mid(subfield(FileName, '\_' ,5),11,2) & ':' & mid(subfield(FileName, '\_' ,5),13,2),'Error'))))>=1,'Date Old','Date Ok') **as** DateAge;

**SELECT** EntryNumber, CASE WHEN BranchCode = '' THEN 'NONE' ELSE UPPER(BranchCode) END AS BranchCode, FileName, EventType, EventDate, SuccessIndicator,

ProcessedErrorMessage, Direction, CAST(XMLData AS varchar) AS XMLData, CASE WHEN ProcessedErrorMessage IS NULL

THEN 'No Error' ELSE ProcessedErrorMessage END AS ErrorMsg, UPPER(LEFT(FileName, 3)) AS Expr1

**FROM** HashTotals WITH(NOLOCK)

WHERE (UPPER(BranchCode) <> 'ALL') AND (UPPER(LEFT(FileName, 3)) <> 'ALL') AND (BranchCode <> '');

*[Temp Last Import]:*

**Load**

Upper(BranchCode) **as** BranchCode,

if(FN\_Transaction\_Type <> 'hastotal' **and** EventType='SBO' **and** xxGood=-1 **and** xxDirection='IN',FileDate) **as** LastUpdate,

if(subfield(FileName, '\_' ,2)='hastotal', mid(subfield(FileName, '\_' ,4),1,4) & '-' & mid(subfield(FileName, '\_' ,4),5,2) & '-' & mid(subfield(FileName, '\_' ,4),7,2) & ' ' & mid(subfield(FileName, '\_' ,4),9,2) & ':' & mid(subfield(FileName, '\_' ,4),11,2) & ':' & mid(subfield(FileName, '\_' ,4),13,2),'') **as** LastPoll

**resident** [Details]

**where** BranchCode <> '';

*[Last Import]:*

**noconcatenate** **Load**

BranchCode,

max(LastUpdate) **as** LastUpdate,

max(LastPoll) **as** LastPoll

**resident** [Temp Last Import]

**where** BranchCode <> ''

**group** **by** BranchCode;

**Drop** **Table** *[Temp Last Import];*

*[Temp]:*

**SELECT** UPPER(BranchCode) AS BranchCode, FileName, CASE Direction WHEN 'IN' THEN 'IN' WHEN 'OUT' THEN 'OUT' ELSE 'ERROR' END AS Direction,

(SELECT COUNT(FileName) AS Expr1

**FROM** HashTotals AS H2

WHERE (H1.FileName = FileName) AND (EventType = 'YMS')) AS YMSTries,

(SELECT COUNT(FileName) AS Expr1

**FROM** HashTotals AS H2

WHERE (H1.FileName = FileName) AND (EventType = 'YMS') AND (SuccessIndicator = 'False')) AS YMSFail,

(SELECT COUNT(FileName) AS Expr1

**FROM** HashTotals AS H2

WHERE (H1.FileName = FileName) AND (EventType = 'YMS') AND (SuccessIndicator = 'True')) AS YMSPass,

(SELECT COUNT(FileName) AS Expr1

**FROM** HashTotals AS H2

WHERE (H1.FileName = FileName) AND (EventType = 'SBO')) AS SBOTries,

(SELECT COUNT(FileName) AS Expr1

**FROM** HashTotals AS H2

WHERE (H1.FileName = FileName) AND (EventType = 'SBO') AND (SuccessIndicator = 'False')) AS SBOFail,

(SELECT COUNT(FileName) AS Expr1

**FROM** HashTotals AS H2

WHERE (H1.FileName = FileName) AND (EventType = 'SBO') AND (SuccessIndicator = 'True')) AS SBOPass, CASE WHEN

(SELECT COUNT(FileName) AS Expr1

**FROM** HashTotals AS H2

WHERE (H1.FileName = FileName) AND (EventType = 'SBO') AND (SuccessIndicator = 'True')) > 1 THEN 'SBO Multi Pass' ELSE 'Ok' END AS SBOMulti,

(SELECT COUNT(FileName) AS Expr1

**FROM** HashTotals AS H2

WHERE (H1.FileName = FileName) AND (EventType = 'IBOLT')) AS IBOLTTries,

(SELECT COUNT(FileName) AS Expr1

**FROM** HashTotals AS H2

WHERE (H1.FileName = FileName) AND (EventType = 'IBOLT') AND (SuccessIndicator = 'False')) AS IBOLTFail,

(SELECT COUNT(FileName) AS Expr1

**FROM** HashTotals AS H2

WHERE (H1.FileName = FileName) AND (EventType = 'IBOLT') AND (SuccessIndicator = 'True')) AS IBOLTPass, UPPER(LEFT(FileName, 3)) AS BrExtract

**FROM** HashTotals AS H1 WITH(NOLOCK)

GROUP BY UPPER(BranchCode), FileName, Direction, UPPER(LEFT(FileName, 3))

HAVING (UPPER(BranchCode) <> 'ALL') AND (UPPER(LEFT(FileName, 3)) <> 'ALL') AND (UPPER(BranchCode) IS NOT NULL) AND (UPPER(BranchCode) <> '');

**left** **join**

**Load**

FileName,

lower(UserName) **as** Fix\_Name,

Comment **as** Fix\_Comments,

LastEdit **as** Fix\_DateTime,

if(YMS=1,'Yes','No') **as** YMSFixed,

if(IBOLT=1,'Yes','No') **as** IBOLTFixed,

if(SBO=1,'Yes','No') **as** SBOFixed,

AssignedTo **as** [Assigned To],

Status **as** [Fix Status],

StartDate **as** [Fix\_StartDate],

if(Status<>'04. Closed',Interval(Now()-StartDate, 'D hh:mm:ss'),Interval(LastEdit-StartDate, 'D hh:mm:ss') ) **as** [Fix\_Duration],

iBoltErrorCode **as** [Error Code],

FixType;

**SELECT** SAPFix.FileName, SAPFix.UserName, SAPFix.Comment, SAPFix.LastEdit, SAPFix.YMS, SAPFix.IBOLT, SAPFix.SBO, SAPFix.FixType, SAPFix.BranchComment,

SAPFix.StartDate, SAPFix.HODate, SAPFix.BranchDate, SAPFix.BranchFixType, SAPFix.CloseDate, RIGHT('00' + CAST(SAPFixStatus.StepNo AS varchar), 2)

+ '. ' + SAPFixStatus.Status AS Status, SAPFixUsers.UserName AS AssignedTo, RIGHT('0000' + CAST(ErrorCodes.iBoltErrorCode AS varchar), 3)

+ ') ' + ErrorCodes.Description AS iBoltErrorCode

**FROM** SAPFix WITH(NOLOCK) INNER JOIN

ErrorCodes WITH(NOLOCK) ON SAPFix.iBoltErrorCode = ErrorCodes.iBoltErrorCode INNER JOIN

SAPFixStatus WITH(NOLOCK) ON SAPFix.Status = SAPFixStatus.AutoID INNER JOIN

SAPFixUsers WITH(NOLOCK) ON SAPFix.AssignedTo = SAPFixUsers.AutoID

WHERE (SAPFix.FixType = 'iBolt');

*[Hash Totals]:*

**noconcatenate**

**load**

BranchCode **as** xBranchCode,

FileName,

Direction,

YMSTries,

YMSFail,

YMSPass,

SBOTries,

SBOFail,

SBOPass,

SBOMulti,

IBOLTTries,

IBOLTFail,

IBOLTPass,

Fix\_Name,

Fix\_Comments,

Fix\_DateTime,

[Fix Status],

[Fix\_Duration],

SBOFixed,

[Assigned To],

[Error Code],

if(YMSPass=1,'N/A',if(YMSPass=0 **and** Direction='IN','Unconfirmed IN',if(YMSPass=0 **and** Direction='OUT','Unconfirmed OUT',if(YMSPass>1 **and** YMSFixed<>'Yes','YMS Error','YMS Fixed')))) **as** YMSStatus,

if(IBOLTPass=1,'N/A',if(IBOLTPass>1,'IBOLT Multi',if(IBOLTPass=0 **and** IBOLTFixed<>'Yes','IBOLT Error','IBOLT Fixed'))) **as** IBOLTStatus,

if(SBOPass=1,'N/A',if(SBOPass>1 **and** SBOFixed<>'Yes','SBO Error',if(SBOPass<>1 **and** SBOFixed<>'Yes' **and** SBOFail>=5,'SBO Error',if(SBOPass<>1 **and** SBOFixed<>'Yes' **and** SBOFail<5 **and** SBOFail>0,'SBO Retry',if(SBOPass=0 **and** SBOFixed<>'Yes','SBO Missing','SBO Fixed'))))) **as** SBOStatus,

if(YMSPass=0 **or** SBOPass=0 **or** IBOLTPass=0, 'Error', if(YMSPass=1 **and** SBOPass=1 **and** IBOLTPass=1,'Single Pass','Multi Pass')) **as** Status

**resident** [Temp]

**WHERE** BranchCode <>'';

**drop** **table** *[Temp];*

**CONNECT** **TO** [Provider=SQLOLEDB.1;Persist Security Info=True;User ID=Qlikview;Initial Catalog=RECLAM\_HO\_PRODUCTION;Data Source=recsap;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=RECQVIEW;Use Encryption for Data=False;Tag with column collation when possible=False] (XPassword is xxx);

**concatenate** (Details)

**load** \*,

'No' **as** Reconciled,

if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines) **as** YMSTicketNo;

**SELECT** 'SAP' AS Source, 'AR Invoice' AS [Document], OINV.DocNum AS [Doc Num], CASE WHEN DocStatus = 'O' THEN 'Open' ELSE 'Closed' END AS [Doc Status],

LTRIM(STR(OINV.ObjType)) + N'-' + LTRIM(STR(OINV.TransId)) AS xxTransLink, OINV.TransId AS TransLink, OINV.CardCode AS [BP Code],

-', INV1.U\_ITN\_LI)

-', INV1.U\_ITN\_LI) - 1) ELSE INV1.U\_ITN\_LI END AS YMSTicketNoLines, CASE WHEN charindex('-', OINV.U\_ITN)

-', OINV.U\_ITN) - 1) ELSE OINV.U\_ITN END AS YMSTicketNoHeader,

AR Credit Memo' WHEN - 1 THEN 'No Target Document' ELSE 'Unknown' END AS [Target Document], INV1.FinncPriod,

CASE BaseType WHEN 15 THEN 'Delivery' WHEN - 1 THEN 'No Base Document' ELSE 'Unknown' END AS [Base Document], UPPER(SUBSTRING(OINV.CardCode, 1,

3)) AS BranchCode, 'Sale AR' AS [Recon Type], OINV.OwnerCode, CASE WHEN oinv.OwnerCode = 39 OR oinv.OwnerCode = 40 THEN 'Auto iBolt' ELSE 'Other' END AS [YMS Ticket Type]

**FROM** OINV WITH(NOLOCK) INNER JOIN

INV1 WITH(NOLOCK) ON OINV.DocEntry = INV1.DocEntry INNER JOIN

OVTG WITH(NOLOCK) ON INV1.VatGroup = OVTG.Code

WHERE (INV1.U\_ITN\_LI IS NOT NULL) OR (OINV.U\_ITN IS NOT NULL);

**Concatenate** (Details)

**load** \*,

'No' **as** Reconciled,

if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines) **as** YMSTicketNo;

**SELECT** 'SAP' AS Source, 'Delivery' AS [Document], ODLN.DocNum AS [Doc Num], CASE WHEN DocStatus = 'O' THEN 'Open' ELSE 'Closed' END AS [Doc Status],

LTRIM(STR(ODLN.ObjType)) + N'-' + LTRIM(STR(ODLN.TransId)) AS TransLink, ODLN.CardCode AS [BP Code], DLN1.ItemCode AS [Item Code],

-', DLN1.U\_ITN\_LI)

-', DLN1.U\_ITN\_LI) - 1) ELSE DLN1.U\_ITN\_LI END AS YMSTicketNoLines, CASE WHEN charindex('-', ODLN.U\_ITN)

-', ODLN.U\_ITN) - 1) ELSE ODLN.U\_ITN END AS YMSTicketNoHeader,

No Target Document' ELSE 'Unknown' END AS [Target Document], DLN1.FinncPriod, 'Sale Delivery' as [Recon Type],ODLN.OwnerCode,

CASE WHEN ODLN.OwnerCode = 39 OR ODLN.OwnerCode = 40 THEN 'Auto iBolt' ELSE 'Other' END AS [YMS Ticket Type],

CASE TargetType WHEN 13 THEN 'AR Invoice' WHEN 16 THEN 'Returns' WHEN - 1 THEN 'No Base Document' ELSE 'Unknown' END AS [Base Document], Upper(SUBSTRING(ODLN.CardCode, 1, 3)) AS [BranchCode]

**FROM** ODLN WITH(NOLOCK) INNER JOIN

DLN1 WITH(NOLOCK) ON ODLN.DocEntry = DLN1.DocEntry INNER JOIN

OVTG WITH(NOLOCK) ON DLN1.VatGroup = OVTG.Code

WHERE (DLN1.U\_ITN\_LI IS NOT NULL) OR (ODLN.U\_ITN IS NOT NULL);

**Concatenate** (Details)

**load** \*,

'No' **as** Reconciled,

if (Right(if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines),1)='R',mid(if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines),1, len(if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines))-1),if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines)) **as** YMSTicketNo;

**SELECT** 'SAP' AS Source, 'Returns' AS [Document], ORDN.DocNum AS [Doc Num], CASE WHEN DocStatus = 'O' THEN 'Open' ELSE 'Closed' END AS [Doc Status],

LTRIM(STR(ORDN.ObjType)) + N'-' + LTRIM(STR(ORDN.TransId)) AS TransLink, ORDN.CardCode AS [BP Code], RDN1.ItemCode AS [Item Code],

-', RDN1.U\_ITN\_LI) > 0 THEN SUBSTRING(RDN1.U\_ITN\_LI, 1, CHARINDEX('-', RDN1.U\_ITN\_LI) - 1)

-', ORDN.U\_ITN) > 0 THEN SUBSTRING(ORDN.U\_ITN, 1,

-', ORDN.U\_ITN) - 1) ELSE ORDN.U\_ITN END AS YMSTicketNoHeader, ORDN.OwnerCode, CASE WHEN ORDN.OwnerCode = 39 OR

Auto iBolt' ELSE 'Other' END AS [YMS Ticket Type],

CASE RDN1.TargetType WHEN 14 THEN 'AR Credit Memo' WHEN - 1 THEN 'No Target Document' ELSE 'Unknown' END AS [Target Document],

'Sale Delivery' AS [Recon Type], CASE RDN1.BaseType WHEN 15 THEN 'Delivery' WHEN - 1 THEN 'No Base Document' ELSE 'Unknown' END AS [Base Document],

UPPER(SUBSTRING(ORDN.CardCode, 1, 3)) AS BranchCode, RDN1.BaseRef, DLN1.U\_ITN\_LI, RDN1.U\_ITN\_LI AS YMSOriginalTicketNo

**FROM** ORDN WITH (NOLOCK) INNER JOIN

RDN1 WITH (NOLOCK) ON ORDN.DocEntry = RDN1.DocEntry INNER JOIN

OVTG WITH (NOLOCK) ON RDN1.VatGroup = OVTG.Code LEFT OUTER JOIN

DLN1 ON RDN1.BaseRef = DLN1.DocEntry AND RDN1.BaseLine = DLN1.LineNum

WHERE (RDN1.U\_ITN\_LI IS NOT NULL) OR

(ORDN.U\_ITN IS NOT NULL);

**Concatenate** (Details)

**load** \*,

'No' **as** Reconciled,

if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines) **as** YMSTicketNo;

**SELECT** 'SAP' AS Source, 'AR Credit Memo' AS [Document], ORIN.DocNum AS [Doc Num], CASE WHEN DocStatus = 'O' THEN 'Open' ELSE 'Closed' END AS [Doc Status],

LTRIM(STR(ORIN.ObjType)) + N'-' + LTRIM(STR(ORIN.TransId)) AS xxTransLink, ORIN.TransId AS TransLink, ORIN.CardCode AS [BP Code],

-', RIN1.U\_ITN\_LI)

-', RIN1.U\_ITN\_LI) - 1) ELSE RIN1.U\_ITN\_LI END AS YMSTicketNoLines, CASE WHEN charindex('-', ORIN.U\_ITN)

-', ORIN.U\_ITN) - 1) ELSE ORIN.U\_ITN END AS YMSTicketNoHeader,

No Target Document' ELSE 'Unknown' END AS [Target Document], 'Sale AR' as [Recon Type],ORIN.OwnerCode,

CASE WHEN ORIN.OwnerCode = 39 OR ORIN.OwnerCode = 40 THEN 'Auto iBolt' ELSE 'Other' END AS [YMS Ticket Type],

CASE BaseType WHEN 203 THEN 'AR Down Payment' WHEN 13 THEN 'AR Invoice' WHEN - 1 THEN 'No Base Document' ELSE 'Unknown' END AS [Base Document], Upper(SUBSTRING(ORIN.CardCode, 1, 3)) AS [BranchCode]

**FROM** ORIN WITH(NOLOCK) INNER JOIN

RIN1 WITH(NOLOCK) ON ORIN.DocEntry = RIN1.DocEntry INNER JOIN

OVTG WITH(NOLOCK) ON RIN1.VatGroup = OVTG.Code

WHERE (RIN1.U\_ITN\_LI IS NOT NULL) OR (ORIN.U\_ITN IS NOT NULL);

**Concatenate** (Details)

**load** \*,

'No' **as** Reconciled,

if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines) **as** YMSTicketNo;

**SELECT** 'SAP' AS Source, 'Goods Receipt PO' AS [Document], OPDN.DocNum AS [Doc Num], CASE WHEN DocStatus = 'O' THEN 'Open' ELSE 'Closed' END AS [Doc Status],

LTRIM(STR(OPDN.ObjType)) + N'-' + LTRIM(STR(OPDN.TransId)) AS TransLink, OPDN.CardCode AS [BP Code], PDN1.ItemCode AS [Item Code],

-', PDN1.U\_ITN\_LI)

-', PDN1.U\_ITN\_LI) - 1) ELSE PDN1.U\_ITN\_LI END AS YMSTicketNoLines, CASE WHEN charindex('-', OPDN.U\_ITN)

-', OPDN.U\_ITN) - 1) ELSE OPDN.U\_ITN END AS YMSTicketNoHeader, 'Purchase Goods Receive' as [Recon Type],

CASE TargetType WHEN 21 THEN 'Goods Return' WHEN 18 THEN 'AP Invoice' WHEN - 1 THEN 'No Target Document' ELSE 'Unknown' END AS [Target Document],OPDN.OwnerCode,

CASE WHEN OPDN.OwnerCode = 39 OR OPDN.OwnerCode = 40 THEN 'Auto iBolt' ELSE 'Other' END AS [YMS Ticket Type],

CASE BaseType WHEN - 1 THEN 'No Base Document' ELSE 'Unknown' END AS [Base Document], Upper(SUBSTRING(OPDN.CardCode, 1, 3)) AS [BranchCode], PDN1.U\_YMSTranDetailId

**FROM** OPDN WITH(NOLOCK) INNER JOIN

PDN1 WITH(NOLOCK) ON OPDN.DocEntry = PDN1.DocEntry INNER JOIN

OVTG WITH(NOLOCK) ON PDN1.VatGroup = OVTG.Code

WHERE (PDN1.U\_ITN\_LI IS NOT NULL) OR (OPDN.U\_ITN IS NOT NULL);

**Concatenate** (Details)

**load** \*,

'No' **as** Reconciled,

if (Right(if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines),1)='R',mid(if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines),1, len(if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines))-1),if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines)) **as** YMSTicketNo;

**SELECT** 'SAP' AS Source, 'Goods Return' AS [Document], ORPD.DocNum AS [Doc Num], CASE WHEN DocStatus = 'O' THEN 'Open' ELSE 'Closed' END AS [Doc Status],

LTRIM(STR(ORPD.ObjType)) + N'-' + LTRIM(STR(ORPD.TransId)) AS TransLink, ORPD.CardCode AS [BP Code], RPD1.ItemCode AS [Item Code],

-', RPD1.U\_ITN\_LI) > 0 THEN SUBSTRING(RPD1.U\_ITN\_LI, 1, CHARINDEX('-', RPD1.U\_ITN\_LI) - 1)

-', ORPD.U\_ITN) > 0 THEN SUBSTRING(ORPD.U\_ITN, 1,

-', ORPD.U\_ITN) - 1) ELSE ORPD.U\_ITN END AS YMSTicketNoHeader, 'Purchase Goods Receive' AS [Recon Type], ORPD.OwnerCode,

CASE WHEN ORPD.OwnerCode = 39 OR

ORPD.OwnerCode = 40 THEN 'Auto iBolt' ELSE 'Other' END AS [YMS Ticket Type],

CASE RPD1.BaseType WHEN 20 THEN 'Goods Received PO' WHEN - 1 THEN 'No Base Document' ELSE 'Unknown' END AS [Base Document],

RPD1.U\_YMSTranDetailId, CASE RPD1.TargetType WHEN - 1 THEN 'No Target Document' ELSE 'Unknown' END AS [Target Document],

UPPER(SUBSTRING(ORPD.CardCode, 1, 3)) AS BranchCode, PDN1.U\_ITN\_LI, RPD1.BaseRef, RPD1.U\_ITN\_LI AS YMSOriginalTicketNo

**FROM** ORPD WITH (NOLOCK) INNER JOIN

RPD1 WITH (NOLOCK) ON ORPD.DocEntry = RPD1.DocEntry INNER JOIN

OVTG WITH (NOLOCK) ON RPD1.VatGroup = OVTG.Code LEFT OUTER JOIN

PDN1 ON RPD1.BaseLine = PDN1.LineNum AND RPD1.BaseRef = PDN1.DocEntry

WHERE (RPD1.U\_ITN\_LI IS NOT NULL) OR

(ORPD.U\_ITN IS NOT NULL);

**Concatenate** (Details)

**load** \*,

'No' **as** Reconciled,

if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines) **as** YMSTicketNo;

**SELECT** 'SAP' AS Source, 'AP Invoice' AS [Document], OPCH.DocNum AS [Doc Num], CASE WHEN DocStatus = 'O' THEN 'Open' ELSE 'Closed' END AS [Doc Status],

LTRIM(STR(OPCH.ObjType)) + N'-' + LTRIM(STR(OPCH.TransId)) AS xxTransLink, OPCH.TransId AS TransLink, OPCH.CardCode AS [BP Code],

-', PCH1.U\_ITN\_LI)

-', PCH1.U\_ITN\_LI) - 1) ELSE PCH1.U\_ITN\_LI END AS YMSTicketNoLines, CASE WHEN charindex('-', OPCH.U\_ITN)

-', OPCH.U\_ITN) - 1) ELSE OPCH.U\_ITN END AS YMSTicketNoHeader, 'Purchase AP' as [Recon Type],OPCH.OwnerCode,

CASE WHEN OPCH.OwnerCode = 39 OR OPCH.OwnerCode = 40 THEN 'Auto iBolt' ELSE 'Other' END AS [YMS Ticket Type],

CASE BaseType WHEN 22 THEN 'Purchase Order' WHEN 20 THEN 'Goods Received PO' WHEN - 1 THEN 'No Base Document' ELSE 'Unknown' END AS [Base Document],

CASE TargetType WHEN 19 THEN 'AP Credit Memo' WHEN - 1 THEN 'No Target Document' ELSE 'Unknown' END AS [Target Document], Upper(SUBSTRING(OPCH.CardCode, 1, 3)) AS [BranchCode]

**FROM** OPCH WITH(NOLOCK) INNER JOIN

PCH1 WITH(NOLOCK) ON OPCH.DocEntry = PCH1.DocEntry INNER JOIN

OVTG WITH(NOLOCK) ON PCH1.VatGroup = OVTG.Code

WHERE (PCH1.U\_ITN\_LI IS NOT NULL) OR (OPCH.U\_ITN IS NOT NULL);

**Concatenate** (Details)

**load** \*,

'No' **as** Reconciled,

if(isNull(YMSTicketNoLines) **or** YMSTicketNoLines = '' ,YMSTicketNoHeader,YMSTicketNoLines) **as** YMSTicketNo;

**SELECT** 'SAP' AS Source, 'AP Credit Memo' AS [Document], ORPC.DocNum AS [Doc Num], CASE WHEN DocStatus = 'O' THEN 'Open' ELSE 'Closed' END AS [Doc Status],

LTRIM(STR(ORPC.ObjType)) + N'-' + LTRIM(STR(ORPC.TransId)) AS xxTransLink, ORPC.TransId AS TransLink, ORPC.CardCode AS [BP Code],

-', RPC1.U\_ITN\_LI)

-', RPC1.U\_ITN\_LI) - 1) ELSE RPC1.U\_ITN\_LI END AS YMSTicketNoLines, CASE WHEN charindex('-', ORPC.U\_ITN)

-', ORPC.U\_ITN) - 1) ELSE ORPC.U\_ITN END AS YMSTicketNoHeader,

AP Invoice' WHEN - 1 THEN 'No Base Document' ELSE 'Unknown' END AS [Base Document], 'Purchase AP' as [Recon Type],ORPC.OwnerCode,

CASE WHEN ORPC.OwnerCode = 39 OR ORPC.OwnerCode = 40 THEN 'Auto iBolt' ELSE 'Other' END AS [YMS Ticket Type],

CASE TargetType WHEN - 1 THEN 'No Target Document' ELSE 'Unknown' END AS [Target Document], Upper(SUBSTRING(ORPC.CardCode, 1, 3)) AS [BranchCode]

**FROM** ORPC WITH(NOLOCK) INNER JOIN

RPC1 WITH(NOLOCK) ON ORPC.DocEntry = RPC1.DocEntry INNER JOIN

OVTG WITH(NOLOCK) ON RPC1.VatGroup = OVTG.Code

WHERE (RPC1.U\_ITN\_LI IS NOT NULL) OR (ORPC.U\_ITN IS NOT NULL);

**CONNECT** **TO** [Provider=SQLOLEDB.1;Persist Security Info=True;User ID=sa;Initial Catalog=ibolt31;Data Source=RECIBOLT2;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=RECQVIEW;Use Encryption for Data=False;Tag with column collation when possible=False] (XPassword is IFMcVZFMKA);

**concatenate** (Details)

**LOAD** 'YMS' **as** Source,

UPPER(left(AccountCode,3)) **as** [BranchCode],

UPPER(BranchCode) **as** [Warehouse],

'No' **as** Reconciled,

if(OriginalTicketNo<>'',OriginalTicketNo,TicketNo) **as** YMSTicketNo,

if(Right(if(OriginalTicketNo<>'',OriginalTicketNo,TicketNo),1)='R','Return','Ok') **as** YMSTicketNoType,

DayStart(CreatedDateTime) **as** [Doc Date],

if(TransactionTypeID=1,'YMS Purchase',if(TransactionTypeID=2,'YMS Sales',if(TransactionTypeID=9,'YMS Sales Return',if(TransactionTypeID=11,'YMS Purchase Reversal',if(TransactionTypeID=12,'YMS Sale Reversal'))))) **as** Document,

if(TransactionTypeID=1 **or** TransactionTypeID=11,'Purchase YMS',if(TransactionTypeID=2 **or** TransactionTypeID=9 **or** TransactionTypeID=12,'Sale YMS')) **as** [Recon Type],

TransactionDetailID **as** YMSTransactionDetailID,

AccountCode **as** [BP Code],

if(AccountTypeID=1,'DPS',if(AccountTypeID=2,'CPS',if(AccountTypeID=3,'Casual'))) **as** YMSAccountTypeID,

ProductCode **as** [Item Code],

if(Adjusted = -1,if(DayStart(CreatedDateTime)<'2099-03-01','Yes','No')) **as** [History Import],

if(Adjusted = -1,if(DayStart(CreatedDateTime)<'2099-03-01',

if(TransactionTypeID=1,NettWeight,

if(TransactionTypeID=2,NettWeight,

if(TransactionTypeID=9,NettWeight\*-1,

if(TransactionTypeID=11,NettWeight\*-1,

if(TransactionTypeID=12,NettWeight\*-1))))),0),

if(TransactionTypeID=1,NettWeight,

if(TransactionTypeID=2,NettWeight,

if(TransactionTypeID=9,NettWeight\*-1,

if(TransactionTypeID=11,NettWeight\*-1,

if(TransactionTypeID=12,NettWeight\*-1)))))) **as** [Doc Qty],

if(TransactionTypeID=1,ValueInclVat,if(TransactionTypeID=2,ValueInclVat,if(TransactionTypeID=9,ValueInclVat\*-1,if(TransactionTypeID=11,ValueInclVat\*-1,if(TransactionTypeID=12,ValueInclVat\*-1))))) **as** [Doc Line Total],

if(Adjusted = -1,'Yes','No') **as** YMSAdjusted,

Reference **as** YMSReference,

TicketNo **as** YMSTempTicketNo,

'Closed' **as** [Doc Status],

'Auto iBolt' **as** [YMS Ticket Type],

TicketNo **as** [Doc Num],

OriginalTicketNo **as** YMSOriginalTicketNo,

YMSReconID **as** YMSReconID,

if(Adjusted = -1,'Yes','No') & if(OriginalTicketNo<>'',OriginalTicketNo,TicketNo) & CreatedDateTime **as** GroupingData;

**SQL** **SELECT** \*

**FROM** YMSRecon WITH(NOLOCK);

*[Final Details]:*

**noconcatenate**

**load** \*,

if(isnum(mid(YMSTicketNo,4,1)),if([YMS Ticket Type]='Auto iBolt','Batch','Legacy'),'Single') **as** [YMS Ticket Group]

**resident** Details

**where** (((**not** isnum(mid(YMSTicketNo,1,1))) **and** (**not** isnum(mid(YMSTicketNo,2,1))) **and** (**not** isnum(mid(YMSTicketNo,3,1)))) **and** (index(YMSTicketNo,' ')=0) **and** (index(YMSTicketNo,'.')=0) **and** (index(YMSTicketNo,'\')=0) **and** (index(YMSTicketNo,'/')=0) **and** (index(YMSTicketNo,':')=0) **and** (len(trim(YMSTicketNo))> 3) **and** (isnum(right(rtrim(YMSTicketNo),1)))) **or** (len(FileName)>0);

**drop** **table** *Details;*

*[TempSummary]:*

**load** **distinct** YMSTicketNo,

[YMS Ticket Group] **as** [Recon Ticket Group]

**resident** [Final Details];

**left** **join**

**load** sum([Doc Qty]) / Count(**distinct** [GroupingData]) **as** [Recon Purchase YMS Qty],

Count(**distinct** [GroupingData]) **as** [Recon Purchase YMS Count],

YMSTicketNo

**resident** [Final Details]

**where** [Recon Type]='Purchase YMS'

**group** **by** YMSTicketNo;

**left** **join**

**load** sum([Doc Qty]) / Count(**distinct** [GroupingData]) **as** [Recon Sale YMS Qty],

Count(**distinct** [GroupingData]) **as** [Recon Sale YMS Count],

YMSTicketNo

**resident** [Final Details]

**where** [Recon Type]='Sale YMS'

**group** **by** YMSTicketNo;

**left** **join**

**load** sum([Doc Qty]) **as** [Recon Sale Delivery Qty],

if(MaxString(if(Document='Delivery', [Doc Status])) = 'Closed','Closed' ,'Open') **as** [Recon Open Del],

Count(**distinct** [Doc Num]) **as** [Recon Sale Del Count],

YMSTicketNo

**resident** [Final Details]

**where** [Recon Type]='Sale Delivery'

**group** **by** YMSTicketNo;

**left** **join**

**load** sum([Doc Qty]) **as** [Recon Sale AR Qty],

YMSTicketNo

**resident** [Final Details]

**where** [Recon Type]='Sale AR'

**group** **by** YMSTicketNo;

**left** **join**

**load** sum([Doc Qty]) **as** [Recon Purchase Goods Receive Qty],

if(MaxString(if(Document='Goods Receipt PO', [Doc Status])) = 'Closed','Closed' ,'Open') **as** [Recon Open GRV],

Count(**distinct** [Doc Num]) **as** [Recon Purchase GRV Count],

YMSTicketNo

**resident** [Final Details]

**where** [Recon Type]='Purchase Goods Receive'

**group** **by** YMSTicketNo;

**left** **join**

**load** sum([Doc Qty]) **as** [Recon Purchase AP Qty],

YMSTicketNo

**resident** [Final Details]

**where** [Recon Type]='Purchase AP'

**group** **by** YMSTicketNo;

**left** **join**

**Load**

FileName **as** YMSTicketNo,

if(Status='03. Closed - Acceptable Condition','Accepted - ','') **as** [Recon Accepted],

lower(UserName) **as** [Recon Fix Name],

Comment **as** [Recon Fix Comments],

LastEdit **as** [Recon Fix DateTime],

ReconAssignedTo **as** [Recon Assigned To],

ReconError **as** [Recon Error Code],

Status **as** [Recon Fix Status],

StartDate **as** [Recon StartDate],

if(Status<>'04. Closed',Interval(Now()-StartDate, 'D hh:mm:ss'),Interval(LastEdit-StartDate, 'D hh:mm:ss') ) **as** [Recon Fix Duration],

FixType **as** [Recon Fix Type];

**SELECT** SAPFix.FileName, SAPFix.UserName, SAPFix.Comment, SAPFix.LastEdit, SAPFix.FixType, SAPFix.StartDate, RIGHT('00' + CAST(SAPFixStatus.StepNo AS varchar), 2)

+ '. ' + SAPFixStatus.Status AS Status, SAPFix.ReconAssignedTo, SAPFix.ReconError

**FROM** SAPFix WITH(NOLOCK) LEFT OUTER JOIN

SAPFixStatus ON SAPFix.Status = SAPFixStatus.AutoID

WHERE (SAPFix.FixType <> 'iBolt');

*[LowDoc]:*

**noconcatenate**

**load** [Document] & minstring([Doc Num]) **as** [Recon Low Doc],

count(**distinct** [Doc Num]) **as** [Recon Low Doc Count],

'X' **as** Base,

YMSTicketNo

**resident** [Final Details]

**where** [Document]='Goods Receipt PO' **or** [Document]='Delivery'

**group** **by** YMSTicketNo,[Recon Type],[Document];

**left** **join**

**load** [Document] & [Doc Num] **as** [Recon Low Doc],

sum([Doc Qty]) **as** [Recon Low Qty],

YMSTicketNo

**resident** [Final Details]

**where** [Document]='Goods Receipt PO' **or** [Document]='Delivery'

**group** **by** YMSTicketNo,[Recon Type],[Document],[Doc Num];

**left** **join**(*[TempSummary]*)

**load**

YMSTicketNo,

[Recon Low Doc],

[Recon Low Doc Count],

[Recon Low Qty]

**resident** [LowDoc] **where** Base = 'X';

*[SummaryFill]:*

**noconcatenate**

**load**

[YMSTicketNo],

if(isnull([Recon Purchase YMS Qty]),0,[Recon Purchase YMS Qty]) **as** [Recon Purchase YMS Qty],

if(isnull([Recon Purchase YMS Count]),0,[Recon Purchase YMS Count]) **as** [Recon Purchase YMS Count],

if(isnull([Recon Sale YMS Qty]),0,[Recon Sale YMS Qty]) **as** [Recon Sale YMS Qty],

if(isnull([Recon Sale YMS Count]),0,[Recon Sale YMS Count]) **as** [Recon Sale YMS Count],

if(isnull([Recon Sale Delivery Qty]),0,[Recon Sale Delivery Qty]) **as** [Recon Sale Delivery Qty],

if(isnull([Recon Open Del]),0,[Recon Open Del]) **as** [Recon Open Del],

if(isnull([Recon Sale Del Count]),0,[Recon Sale Del Count]) **as** [Recon Sale Del Count],

if(isnull([Recon Sale AR Qty]),0,[Recon Sale AR Qty]) **as** [Recon Sale AR Qty],

if(isnull([Recon Purchase Goods Receive Qty]),0,[Recon Purchase Goods Receive Qty]) **as** [Recon Purchase Goods Receive Qty],

if(isnull([Recon Open GRV]),0,[Recon Open GRV]) **as** [Recon Open GRV],

if(isnull([Recon Purchase GRV Count]),0,[Recon Purchase GRV Count]) **as** [Recon Purchase GRV Count],

if(isnull([Recon Purchase AP Qty]),0,[Recon Purchase AP Qty]) **as** [Recon Purchase AP Qty],

[Recon Accepted],

[Recon Fix Name],

[Recon Fix Comments],

[Recon Fix DateTime],

[Recon Assigned To],

[Recon Error Code],

[Recon Fix Status],

[Recon StartDate],

[Recon Fix Duration],

[Recon Fix Type],

[Recon Ticket Group],

if(isnull([Recon Low Doc]),0,[Recon Low Doc]) **as** [Recon Low Doc],

if(isnull([Recon Low Doc Count]),0,[Recon Low Doc Count]) **as** [Recon Low Doc Count],

if(isnull([Recon Low Qty]),0,[Recon Low Qty]) **as** [Recon Low Qty]

**resident** TempSummary;

*[Summary]:*

**noconcatenate**

**load** \*,

if([Recon Ticket Group]='Legacy', [Recon Accepted]&'Pass - Legacy transaction',

if(([Recon Purchase Goods Receive Qty]/[Recon Purchase YMS Qty])>=(***$(IntegrationTolerance)***/100),[Recon Accepted]&'Fail - Int. Duplicate',

if(([Recon Sale Delivery Qty]/[Recon Sale YMS Qty])>=(***$(IntegrationTolerance)***/100),[Recon Accepted]&'Fail - Int. Duplicate',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty]= 0 **and** [Recon Purchase AP Qty]=0 **and** [Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty]=0 **and** [Recon Sale AR Qty]=0 ,[Recon Accepted]&'AP Pass - Zero Transaction',

if([Recon Purchase YMS Qty]=[Recon Purchase Goods Receive Qty] **and** [Recon Purchase YMS Qty]= [Recon Purchase AP Qty] **and** [Recon Purchase YMS Qty]<>0,[Recon Accepted]&'AP Pass',

if([Recon Purchase YMS Qty]=[Recon Low Qty] **and** [Recon Purchase Goods Receive Qty]= [Recon Purchase AP Qty] **and** [Recon Purchase YMS Qty]<>0 **and** [Recon Purchase AP Qty]<>0,[Recon Accepted]&'AP Pass - Re-Receipt',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]>=2, [Recon Accepted]&'AP Fail - Transaction Cancelled in SAP only',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0 **and** [Recon Purchase AP Qty] = 0 **and** [Recon Ticket Group] = 'Single', [Recon Accepted]&'AP Fail - (Single) YMS Exists - No SAP Transactions - Int. Issue',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0 **and** [Recon Purchase AP Qty] = 0 **and** [Recon Ticket Group] = 'Batch', [Recon Accepted]&'AP Fail - (Batch) YMS Exists - No SAP Transactions - Int. Issue',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0 **and** [Recon Purchase AP Qty] > 0 **and** [Recon Ticket Group] = 'Single', [Recon Accepted]&'AP Fail - YMS Exists - Manual SAP Transactions and Int. Issue',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0 **and** [Recon Purchase AP Qty] > 0 **and** [Recon Ticket Group] = 'Batch' **and** [Recon Purchase YMS Qty]=[Recon Purchase AP Qty], [Recon Accepted]&'AP Pass - YMS Batch',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0 **and** [Recon Purchase AP Qty] > 0, [Recon Accepted]&'AP Fail - YMS does NOT exsist - Manual SAP Transaction',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty] <> 0 **and** [Recon Purchase AP Qty] <> 0, [Recon Accepted]&'AP Fail - YMS does NOT exsist - Manual SAP Transaction',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty] <> 0 **and** [Recon Purchase AP Qty] = 0, [Recon Accepted]&'AP Fail - YMS does NOT exsist - Manual SAP Transaction',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0, [Recon Accepted]&'AP Fail - YMS Exists - No GR in SAP - Could be in transit',

if([Recon Purchase Goods Receive Qty] <> [Recon Purchase AP Qty] **and** [Recon Open GRV] ='Open', [Recon Accepted]&'AP Invoicing not complete',

if([Recon Purchase Goods Receive Qty] <> [Recon Purchase AP Qty] **and** [Recon Purchase Goods Receive Qty]=0 **and** [Recon Purchase AP Qty]>0,[Recon Accepted]&'AP Goods Receipt = 0 and Invoice exists',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase YMS Count]>=2 **and** [Recon Purchase GRV Count]>=2, [Recon Accepted]&'AP Pass - Transaction Cancelled in YMS and SAP',

if([Recon Purchase Goods Receive Qty] <> [Recon Purchase AP Qty] ,[Recon Accepted]&'AP Invoice Out of Balance with Goods Receipt',

if([Recon Purchase YMS Qty] = 0 **and** [Recon Purchase Goods Receive Qty] <> 0 **and** [Recon Purchase YMS Count]=0 , [Recon Accepted]&'AP YMS Missing - No YMS Records - SAP Goods Receipt exists',

if([Recon Purchase YMS Qty] = 0 **and** [Recon Purchase Goods Receive Qty] <> 0 **and** [Recon Purchase YMS Count]>=1 , [Recon Accepted]&'AP YMS Missing - YMS Records Found - SAP Goods Receipt exists',

if([Recon Purchase YMS Qty] = 0 **and** [Recon Purchase AP Qty] <> 0 **and** [Recon Purchase YMS Count]=0 , [Recon Accepted]&'AP YMS Missing - No YMS Records - SAP Invoice exists',

if([Recon Purchase YMS Qty] = 0 **and** [Recon Purchase AP Qty] <> 0 **and** [Recon Purchase YMS Count]>=1 , [Recon Accepted]&'AP YMS Missing - YMS Records Found - SAP Invoice exists',

if([Recon Purchase YMS Qty] <> [Recon Purchase Goods Receive Qty] **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]>=2,[Recon Accepted]&'AP YMS <> SAP Goods Receipt and SAP has been Returned',

if([Recon Purchase Goods Receive Qty] = [Recon Purchase AP Qty] **and** [Recon Purchase YMS Qty] <> [Recon Purchase Goods Receive Qty],[Recon Accepted]&'AP YMS SAP Variance',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty]= 0 **and** [Recon Purchase AP Qty]=0 **and** [Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty]=0 **and** [Recon Sale AR Qty]=0 ,[Recon Accepted]&'AR Pass - Zero Transaction',

if([Recon Sale YMS Qty]=[Recon Sale Delivery Qty] **and** [Recon Sale Delivery Qty] = [Recon Sale AR Qty] **and** [Recon Sale YMS Qty]<>0,[Recon Accepted]&'AR Pass',

if([Recon Sale YMS Qty]=[Recon Low Qty] **and** [Recon Sale Delivery Qty]= [Recon Sale AR Qty] **and** [Recon Sale YMS Qty]<>0 **and** [Recon Sale AR Qty]<>0,[Recon Accepted]&'AR Pass - Re-Delivery',

if([Recon Sale YMS Qty]>0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale Del Count]>=2, [Recon Accepted]&'AR Fail - Transaction Cancelled in SAP only',

if([Recon Sale YMS Qty]>0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale Del Count]=0 **and** [Recon Sale AR Qty] = 0, [Recon Accepted]&'AR Fail - YMS Exists - No SAP Transactions - Int. Issue',

if([Recon Sale YMS Qty]>0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale Del Count]=0 **and** [Recon Sale AR Qty] > 0, [Recon Accepted]&'AR Fail - YMS Exists - Manual SAP Transactions and Int. Issue',

if([Recon Sale YMS Qty]>0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale Del Count]=0, [Recon Accepted]&'AR Fail - YMS Exists - No Del in SAP - Could be in transit',

if([Recon Sale Delivery Qty] <> [Recon Sale AR Qty] **and** [Recon Open Del] ='Open', [Recon Accepted]&'AR Invoicing not complete',

if([Recon Sale Delivery Qty] <> [Recon Sale AR Qty] **and** [Recon Sale Delivery Qty]=0 **and** [Recon Sale AR Qty]>0,[Recon Accepted]&'AR Delivery = 0 and Invoice exists',

if([Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale YMS Count]>=2 **and** [Recon Sale Del Count]>=2, [Recon Accepted]&'AR Pass - Transaction Cancelled in YMS and SAP',

if([Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale YMS Count]=0 **and** [Recon Sale AR Qty] > 0 , [Recon Accepted]&'AR Fail - YMS does NOT exsist - Manual SAP Transaction',

if([Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty] <> 0 **and** [Recon Sale AR Qty] <> 0 , [Recon Accepted]&'AR Fail - YMS does NOT exsist - Manual SAP Transaction',

if([Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty] <> 0 **and** [Recon Sale AR Qty] = 0 , [Recon Accepted]&'AR Fail - YMS does NOT exsist - Manual SAP Transaction',

if([Recon Sale Delivery Qty] <> [Recon Sale AR Qty] ,[Recon Accepted]&'AR Invoice Out of Balance with Delivery',

if([Recon Sale YMS Qty] = 0 **and** [Recon Sale Delivery Qty] <> 0 **and** [Recon Sale YMS Qty]=0 , [Recon Accepted]&'AR YMS Missing - No YMS Records - SAP Delivery exists',

if([Recon Sale YMS Qty] = 0 **and** [Recon Sale Delivery Qty] <> 0 **and** [Recon Sale YMS Count]>=1 , [Recon Accepted]&'AR YMS Missing - YMS Records Found - SAP Delivery exists',

if([Recon Sale YMS Qty] = 0 **and** [Recon Sale AR Qty] <> 0 **and** [Recon Sale YMS Count]=0 , [Recon Accepted]&'AR YMS Missing - No YMS Records - SAP Invoice exists',

if([Recon Sale YMS Qty] = 0 **and** [Recon Sale AR Qty] <> 0 **and** [Recon Sale YMS Count]>=1 , [Recon Accepted]&'AR YMS Missing - YMS Records Found - SAP Invoice exists',

if([Recon Sale YMS Qty] <> [Recon Sale Delivery Qty] **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale Del Count]>=2,[Recon Accepted]&'AR YMS <> SAP Delivery and SAP has been Returned',

if([Recon Sale Delivery Qty] = [Recon Sale AR Qty] **and** [Recon Sale YMS Qty] <> [Recon Sale Delivery Qty],[Recon Accepted]&'AR YMS SAP Variance'

))))))))))))))))))))))))))))))))))))))))))))) **as** [Recon Status],

if([Recon Accepted]='Accepted - ', 'Pass',

if([Recon Ticket Group]='Legacy', 'Pass',

if(([Recon Purchase Goods Receive Qty]/[Recon Purchase YMS Qty])>=(***$(IntegrationTolerance)***/100),'Fail',

if(([Recon Sale Delivery Qty]/[Recon Sale YMS Qty])>=(***$(IntegrationTolerance)***/100),'Fail',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty]= 0 **and** [Recon Purchase AP Qty]=0 **and** [Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty]=0 **and** [Recon Sale AR Qty]=0 ,'Pass',

if([Recon Purchase YMS Qty]=[Recon Purchase Goods Receive Qty] **and** [Recon Purchase YMS Qty]= [Recon Purchase AP Qty] **and** [Recon Purchase YMS Qty]<>0 **and** [Recon Purchase Goods Receive Qty]<>0 **and** [Recon Purchase AP Qty]<>0 ,'Pass',

if([Recon Purchase YMS Qty]=[Recon Low Qty] **and** [Recon Purchase Goods Receive Qty]= [Recon Purchase AP Qty] **and** [Recon Purchase YMS Qty]<>0 **and** [Recon Purchase AP Qty]<>0,'Pass',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]>=2, 'Fail',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0 **and** [Recon Purchase AP Qty] = 0, 'Fail',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0 **and** [Recon Purchase AP Qty] > 0 **and** [Recon Ticket Group] = 'Single', 'Fail',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0 **and** [Recon Purchase AP Qty] > 0 **and** [Recon Ticket Group] = 'Batch' **and** [Recon Purchase YMS Qty]=[Recon Purchase AP Qty], 'Pass',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0 **and** [Recon Purchase AP Qty] > 0 , 'Fail',

if([Recon Purchase YMS Qty]>0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]=0, 'Fail',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty] <> 0 **and** [Recon Purchase AP Qty] <> 0 , 'Fail',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty] <> 0 **and** [Recon Purchase AP Qty] = 0 , 'Fail',

if([Recon Purchase Goods Receive Qty] <> [Recon Purchase AP Qty] **and** [Recon Open GRV] ='Open', 'Incomplete',

if([Recon Purchase Goods Receive Qty] <> [Recon Purchase AP Qty] **and** [Recon Purchase Goods Receive Qty]=0 **and** [Recon Purchase AP Qty]>0,'Fail',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase YMS Count]>=2 **and** [Recon Purchase GRV Count]>=2, 'Pass',

if([Recon Purchase Goods Receive Qty] <> [Recon Purchase AP Qty] ,'Fail',

if([Recon Purchase YMS Qty] = 0 **and** [Recon Purchase Goods Receive Qty] <> 0 **and** [Recon Purchase YMS Count]=0 , 'Fail',

if([Recon Purchase YMS Qty] = 0 **and** [Recon Purchase Goods Receive Qty] <> 0 **and** [Recon Purchase YMS Count]>=1 , 'Fail',

if([Recon Purchase YMS Qty] = 0 **and** [Recon Purchase AP Qty] <> 0 **and** [Recon Purchase YMS Count]=0 , 'Fail',

if([Recon Purchase YMS Qty] = 0 **and** [Recon Purchase AP Qty] <> 0 **and** [Recon Purchase YMS Count]>=1 , 'Fail',

if([Recon Purchase YMS Qty] <> [Recon Purchase Goods Receive Qty] **and** [Recon Purchase Goods Receive Qty] = 0 **and** [Recon Purchase GRV Count]>=2,'Fail',

if([Recon Purchase Goods Receive Qty] = [Recon Purchase AP Qty] **and** [Recon Purchase YMS Qty] <> [Recon Purchase Goods Receive Qty],'Fail',

if([Recon Purchase YMS Qty]=0 **and** [Recon Purchase Goods Receive Qty]= 0 **and** [Recon Purchase AP Qty]=0 **and** [Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty]=0 **and** [Recon Sale AR Qty]=0 ,'Pass',

if([Recon Sale YMS Qty]=[Recon Sale Delivery Qty] **and** [Recon Sale YMS Qty]= [Recon Sale AR Qty] **and** [Recon Sale YMS Qty]<>0 **and** [Recon Sale Delivery Qty]<>0 **and** [Recon Sale AR Qty]<>0 ,'Pass',

if([Recon Sale YMS Qty]=[Recon Low Qty] **and** [Recon Sale Delivery Qty]= [Recon Sale AR Qty] **and** [Recon Sale YMS Qty]<>0 **and** [Recon Sale AR Qty]<>0,'Pass',

if([Recon Sale YMS Qty]>0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale Del Count]>=2, 'Fail',

if([Recon Sale YMS Qty]>0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale Del Count]=0 **and** [Recon Sale AR Qty] = 0, 'Fail',

if([Recon Sale YMS Qty]>0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale Del Count]=0 **and** [Recon Sale AR Qty] > 0, 'Fail',

if([Recon Sale YMS Qty]>0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale Del Count]=0, 'Fail',

if([Recon Sale Delivery Qty] <> [Recon Sale AR Qty] **and** [Recon Open Del] ='Open', 'Incomplete',

if([Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty] <> 0 **and** [Recon Sale AR Qty] <> 0, 'Fail',

if([Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty] <> 0 **and** [Recon Sale AR Qty] = 0, 'Fail',

if([Recon Sale Delivery Qty] <> [Recon Sale AR Qty] **and** [Recon Sale Delivery Qty]=0 **and** [Recon Sale AR Qty]>0,'Fail',

if([Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale YMS Count]>=2 **and** [Recon Sale Del Count]>=2, 'Pass',

if([Recon Sale YMS Qty]=0 **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale YMS Count]=0 **and** [Recon Sale AR Qty] > 0 , 'Fail',

if([Recon Sale Delivery Qty] <> [Recon Sale AR Qty] ,'Fail',

if([Recon Sale YMS Qty] = 0 **and** [Recon Sale Delivery Qty] <> 0 **and** [Recon Sale YMS Qty]=0 , 'Fail',

if([Recon Sale YMS Qty] = 0 **and** [Recon Sale Delivery Qty] <> 0 **and** [Recon Sale YMS Count]>=1 , 'Fail',

if([Recon Sale YMS Qty] = 0 **and** [Recon Sale AR Qty] <> 0 **and** [Recon Sale YMS Count]=0 , 'Fail',

if([Recon Sale YMS Qty] = 0 **and** [Recon Sale AR Qty] <> 0 **and** [Recon Sale YMS Count]>=1 , 'Fail',

if([Recon Sale YMS Qty] <> [Recon Sale Delivery Qty] **and** [Recon Sale Delivery Qty] = 0 **and** [Recon Sale Del Count]>=2,'Fail',

if([Recon Sale Delivery Qty] = [Recon Sale AR Qty] **and** [Recon Sale YMS Qty] <> [Recon Sale Delivery Qty],'Fail'

))))))))))))))))))))))))))))))))))))))))))))) **as** [Recon Summary Status]

**resident** SummaryFill;

**Drop** **Table** *TempSummary;*

**Drop** **Table** *SummaryFill;*

**Drop** **Table** *[LowDoc];*

**CONNECT** **TO** [Provider=SQLOLEDB.1;Persist Security Info=True;User ID=Qlikview;Initial Catalog=RECLAM\_HO\_PRODUCTION;Data Source=recsap;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=RECQVIEW;Use Encryption for Data=False;Tag with column collation when possible=False] (XPassword is xxx);

*[Fin Periods]:*

**SELECT** AbsEntry AS FinCode, Code AS [Fin Name], SubNum AS [Fin Period], Category AS [Fin Year], F\_RefDate AS [From Date], T\_RefDate AS [To Date],

CASE WHEN SubNum >= 1 AND SubNum <= 3 THEN 'Q1' WHEN SubNum >= 4 AND SubNum <= 6 THEN 'Q2' WHEN SubNum >= 7 AND

SubNum <= 9 THEN 'Q3' WHEN SubNum >= 10 AND SubNum <= 12 THEN 'Q4' END AS [Fin Quarter], CAST(Code + N'-01' AS datetime)

*[Reporting Fin Period]:*

**SELECT** Code AS [Report Fin Name], CAST(Code + N'-01' AS datetime) AS [Report Fin Date]

*[Doc Owner]:*

**SELECT** empID AS OwnerCode, firstName + N' ' + lastName AS [Owner Name]

*[Final Summary]:*

**noconcatenate**

**load** \*

**resident** Summary

**where** ((**not** isnum(mid(YMSTicketNo,1,1))) **and** (**not** isnum(mid(YMSTicketNo,2,1))) **and** (**not** isnum(mid(YMSTicketNo,3,1)))) **and** (index(YMSTicketNo,' ')=0) **and** (index(YMSTicketNo,'.')=0) **and** (index(YMSTicketNo,'\')=0) **and** (index(YMSTicketNo,'/')=0) **and** (index(YMSTicketNo,':')=0) **and** (len(trim(YMSTicketNo))> 3) **and** (isnum(right(rtrim(YMSTicketNo),1)));

**drop** **table** *Summary;*

**IntervalMatch** ( [Doc Date] ) **select** F\_RefDate AS [From Date], T\_RefDate AS [To Date] **from** OFPR;

**Let** TimeStamp=Now();