TEMPLATE for CWT Standard Stored Procedure

Currently CWT files must be loaded directly into Production.

So, create sp in Production Database.

The reason files are loaded directly into Production is because

I.)

CWT sends data based on their change dates.

If they make a change, for example to ReasonCodes,

then the change may generate a years worth of data causing huge files.

II.)

CWT sends an extra file containing Recordkeys to Delete before importing

which uses the dba.CWTGTEMP table.

\*\* This table must be in the database. (See file to create table)

Therefore, deleting large amounts of data across server takes up processing time and memory.

So, it is better to have this done in Production.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Currently,

A CWTHtlChains table is also used for converting 3 letter chain codes to 2 in CWT sps.

However this table is not maintained very much, because of the use of HNN.

---- 1) Select Edit – Replace

---- Find: DATABASE\_NAME

---- Replace; (Actual database name like Tman503\_Name)

---- 2) Select Edit – Replace

---- Find: sp\_NAME

---- Replace: (Actual sp name to mimic Request Name in caps)

---- 3) Select Edit – Replace

---- Find: 'IATANUM'

---- Replace: Actual IATANUM **surrounded by single quotes** so that other locations with IATAUM do not change such as @IATANUM

USE [DATABASE\_NAME]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[sp\_NAME] Script Date: 3/6/2015 10:21:43 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[sp\_NAME]

@BeginIssueDate datetime,

@EndIssueDate datetime

AS

SET NOCOUNT ON

DECLARE @Iata varchar(50), @ProcName varchar(50), @TransStart datetime

,@MaxImportDt datetime,@FirstInvDate datetime,@LastInvDate datetime

,@IataNum varchar(8)

----- For Logging Prompts Only -------------------------------------

SET @Iata = 'IATANUM'

SET @ProcName = CONVERT(varchar(50),OBJECT\_NAME(@@PROCID))

--------------------------------------------------------------

----- For SP PROMPTS ONLY ----------

SET @IataNum = 'IATANUM’

----- \*Note: CWT sends by Change Date

----- So DO NOT use filter: and StagIH.InvoiceDate BETWEEN @BeginIssueDate and @EndIssueDate

SELECT @MaxImportDt = max(IH.ImportDt)

FROM dba.InvoiceHeader IH

WHERE IH.IataNum = @IataNum

---- and StagIH.InvoiceDate BETWEEN @BeginIssueDate and @EndIssueDate

print @MaxImportDt

SELECT @FirstInvDate = min(invoicedate),@LastInvDate = max(invoicedate)

FROM dba.InvoiceHeader

WHERE importdt = @MaxImportDt

and IataNum = @IataNum

print @FirstInvDate

print @LastInvDate

----------------------------------------------------------------

-- Any and all edits can be logged using sp\_LogProcErrors

-- INSERT a row into the dbo.procedurelogs table using sp....when the procedure starts

-- Define all the parameters needed to log each step of a stored procedure and raise an error if necessary

-- @ProcedureName varchar(50), -- \*\*REQUIRED\*\* Name of the Procedure being run

-- @LogStart datetime, -- \*\*REQUIRED\*\* Create a variable and capture a timestamp at the beginning of a transaction

-- @StepName varchar(50) = NULL, -- \*\*OPTIONAL\*\* Name of the Step in the Stored Procedure (For Ex. 'UPDATE ComRmks' OR 'INSERT Udef'

-- @BeginDate datetime = NULL, -- \*\*OPTIONAL\*\* The BeginIssueDate that is passed to the Parent Procedure

-- @EndDate datetime = NULL, -- \*\*OPTIONAL\*\* The EndIssueDate that is pased to the Parent Procedure

-- @IataNum varchar(50) = NULL, -- \*\*OPTIONAL\*\* The IataNum that is passed to the Parent Procedure

-- @RowCount int, -- \*\*REQUIRED\*\* Total number of affected rows

-- @ERR int) -- \*\*REQUIRED\*\* The Error number, even if it is still 0 (Zero)

--Log Activity

SET @TransStart = getdate()

EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='Stored Procedure Start',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

--------------------------------------------------------------------------------

---- Standard CWT Updates ------

---- Originally 4 standard updates - Pam S eliminated 'CWT hotel chain codes' step on 1/26/2015 per SF 00052838

---- So now only 3 standard CWT updates

SET @TransStart = getdate()

UPDATE Car

SET car.issuedate = ID.issuedate

FROM dba.Car CAR

INNER JOIN dba.InvoiceHeader IH

ON (IH.IataNum = CAR.IataNum

AND IH.ClientCode = CAR.ClientCode

AND IH.RecordKey = CAR.RecordKey

AND IH.InvoiceDate = CAR.InvoiceDate)

INNER JOIN dba.InvoiceDetail ID

ON (ID.Iatanum = Car.Iatanum

and ID.ClientCode = Car.ClientCode

and ID.RecordKey = Car.RecordKey

and ID.SeqNum = Car.SeqNum)

WHERE IH.IMPORTDT = @MaxImportDt

AND IH.IataNum = @IataNum

AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

AND CAR.IataNum = @IataNum

AND CAR.InvoiceDate between @FirstInvDate AND @LastInvDate

AND ID.Iatanum = @IataNum

AND ID.InvoiceDate between @FirstInvDate AND @LastInvDate

AND CAR.IssueDate <> ID.IssueDate

EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='Car to ID.IssueDate',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

SET @TransStart = getdate()

UPDATE HTL

SET HTL.Issuedate = ID.issuedate

from dba.Hotel HTL

INNER JOIN dba.InvoiceHeader IH

ON (IH.IataNum = HTL.IataNum

AND IH.ClientCode = HTL.ClientCode

AND IH.RecordKey = HTL.RecordKey

AND IH.InvoiceDate = HTL.InvoiceDate)

INNER JOIN dba.InvoiceDetail ID

ON (ID.Iatanum = HTL.Iatanum

and ID.ClientCode = HTL.ClientCode

and ID.RecordKey = HTL.RecordKey

and ID.SeqNum = HTL.SeqNum)

WHERE IH.IMPORTDT = @MaxImportDt

AND IH.IataNum = @IataNum

AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

AND HTL.IataNum = @IataNum

AND HTL.InvoiceDate between @FirstInvDate AND @LastInvDate

AND ID.Iatanum = @IataNum

AND ID.InvoiceDate between @FirstInvDate AND @LastInvDate

AND HTL.IssueDate <> ID.IssueDate

EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='HTL to ID.IssueDate',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

---- Need to question Sue and Nina - about if CWT still doing this and where to get data to populate table\*\*\*\*\*

---- Per Sue's comment in SF 00052946 on 12/29/2014:

---- "As for the CWTHtlChains table, this was created a long time ago and really hasn't been maintained much. CWT uses 3 character chain codes and this was used to convert them to the standard 2 characters.

---- Now that HNN is used for most clients, it is probably not necessary."

---- Per SF 00052838 Pam S commented out this step after reviewing and analyzing data after CWTHtlChains table was loaded -See comments in SF 00052838 on 1/26/2015.

-- CWT standard update: CWT hotel chain codes

--SET @TransStart = getdate()

--UPDATE HTL

--SET HTL.HtlChainCode = CWTCH.trxcode,

--HTL.HtlChainName = CWTCH.TRXChainName

--FROM dba.Hotel HTL

--INNER JOIN dba.CWTHtlChains CWTCH

--ON (HTL.HtlChainCode = CWTCH.CWTcode)

--INNER JOIN dba.InvoiceHeader IH

--ON (HTL.Iatanum = IH.Iatanum

-- and HTL.ClientCode = IH.ClientCode

-- and HTL.Recordkey = IH.Recordkey

-- and HTL.InvoiceDate = IH.InvoiceDate)

--WHERE IH.IMPORTDT = @MaxImportDt

-- AND IH.Iatanum = @IataNum

-- AND IH.InvoiceDate between @FirstInvDate and @LastInvDate

-- AND HTL.IataNum = @IataNum

-- AND HTL.InvoiceDate between @FirstInvDate and @LastInvDate

-- AND len(HTL.HtlChainCode) > 2

--EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='CWT hotel chain codes',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

---------------------------------------------------------------------------------

---- Update car chain code ZL1/ZL2 to ZL and update car chain name to National

--------------------------------------------------------------------------------

SET @TransStart = getdate()

UPDATE dba.Car

SET

CarChainName = 'NATIONAL CAR RENTAL'

,CarChainCode = 'ZL'

FROM dba.Car CAR

INNER JOIN dba.InvoiceHeader IH

ON (IH.IataNum = CAR.IataNum

AND IH.ClientCode = CAR.ClientCode

AND IH.RecordKey = CAR.RecordKey

AND IH.InvoiceDate = CAR.InvoiceDate)

WHERE IH.IMPORTDT = @MaxImportDt

AND IH.IataNum = @IataNum

AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

AND CAR.IataNum = @IataNum

AND CAR.InvoiceDate between @FirstInvDate AND @LastInvDate

AND CAR.CarChainName is null

AND CAR.CarDailyRate is not null

AND (CAR.CarChainCode = 'ZL1' or CAR.CarChainCode = 'ZL2')

EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='CWT car chain code ZL1/ZL2 to ZL',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

----End standard 4 updates for CWT

---------------------------------------------------------------------------------

---- BEGIN PREFERRED CAR and AIR Vendors

---------------------------------------------------------------------------------

-----------------------------------

------ Preferred Car Vendor

-----------------------------------

------ For dba.Car

------ based on CarChainCode

--SET @TransStart = getdate()

--update dba.car

--set prefcarind = 'Y'

--FROM dba.Car CAR

--INNER JOIN dba.InvoiceHeader IH

-- ON (IH.IataNum = CAR.IataNum

-- AND IH.ClientCode = CAR.ClientCode

-- AND IH.RecordKey = CAR.RecordKey

-- AND IH.InvoiceDate = CAR.InvoiceDate)

--WHERE IH.IMPORTDT = @MaxImportDt

-- AND IH.IataNum = @IataNum

-- AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND CAR.IataNum = @IataNum

-- AND CAR.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND CAR.CarChainCode in ('')

-- AND CAR.PrefCarInd is null

--EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='Preferred Car Vendor',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

-----------------------------------

------ Preferred Air Vendors

-----------------------------------

------------------------------------------------------------------------------------------

------ For dba.InvoiceDetal

------ based on ID.ValCarrierCode and sometimes IH.OrigCountry

------ So will need to either add a Case when statement or filter depending on complexity

------------------------------------------------------------------------------------------

--SET @TransStart = getdate()

--UPDATE dba.InvoiceDetail

--SET PrefTktInd =

--FROM dba.InvoiceDetail ID

--INNER JOIN dba.InvoiceHeader IH

-- ON (IH.IataNum = ID.IataNum

-- AND IH.ClientCode = ID.ClientCode

-- AND IH.RecordKey = ID.RecordKey

-- AND IH.InvoiceDate = ID.InvoiceDate)

--INNER JOIN dba.Country CTRY

-- On (IH.OrigCountry = CTRY.CtryCode)

--WHERE IH.IMPORTDT = @MaxImportDt

-- AND IH.IataNum = @IataNum

-- AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND ID.IataNum = @IataNum

-- AND ID.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND ID.PrefTktInd is null

--EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='ID.PrefTktInd- AIR Vendor',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

------ For dba.Transeg

------ based on TS.SegmentCarrierCode and sometimes IH.OrigCountry

------ So will need to either add a Case when statement or filter depending on complexity

--SET @TransStart = getdate()

--UPDATE dba.Transeg

--SET PrefAirInd =

--FROM dba.TranSeg TS

--INNER JOIN dba.InvoiceHeader IH

-- ON (IH.IataNum = TS.IataNum

-- AND IH.ClientCode = TS.ClientCode

-- AND IH.RecordKey = TS.RecordKey

-- AND IH.InvoiceDate = TS.InvoiceDate)

--INNER JOIN dba.Country CTRY

-- On (IH.OrigCountry = CTRY.CtryCode)

--WHERE IH.IMPORTDT = @MaxImportDt

-- AND IH.IataNum = @IataNum

-- AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND TS.IataNum = @IataNum

-- AND TS.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND TS.PrefAirInd is null

--EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='TS.PrefAIRInd',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

---------------------------------------------------------------------------------

---- END PREFERRED CAR and AIR Vendors

---------------------------------------------------------------------------------

-----------------------------

---- CLASS TO CABIN

-----------------------------

----------------------------------------------------------------------------------------------

---- Before ComRmks Updates for Text14 Highest Class

----------------------------------------------------------------------------------------------

----- From InvoiceDetail into Production Class to Cabin:

SET @TransStart = getdate()

INSERT INTO dba.classtocabin

SELECT distinct substring(ID.valCarrierCode,1,3), substring(ID.servicecategory,1,1), 'ECONOMY',ID.InternationalInd,'Y',NULL,NULL

FROM dba.InvoiceDetail ID

INNER JOIN dba.InvoiceHeader IH

ON (IH.IataNum = ID.IataNum

AND IH.ClientCode = ID.ClientCode

AND IH.RecordKey = ID.RecordKey

AND IH.InvoiceDate = ID.InvoiceDate)

WHERE IH.IMPORTDT = @MaxImportDt

AND IH.IataNum = @IataNum

AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

AND ID.IataNum = @IataNum

AND ID.InvoiceDate between @FirstInvDate AND @LastInvDate

AND ID.ValCarrierCode IS NOT NULL

AND ID.Servicecategory IS NOT NULL

AND ID.VendorType in ('BSP','NONBSP','RAIL')

AND substring(ID.ValCarrierCode,1,3)+substring(ID.servicecategory,1,1)+ID.InternationalInd

NOT IN (SELECT DISTINCT CarrierCode+ClassOfService+InternationalInd

FROM .DBA.ClassToCabin)

EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='ClassToCabin FROM InvoiceDetalil',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

-----From Transeg into Production Class to Cabin:

SET @TransStart = getdate()

INSERT INTO .dba.classtocabin

SELECT distinct substring(TS.SegmentCarrierCode,1,3), substring(TS.ClassOfService,1,1), 'ECONOMY',TS.segInternationalInd,'Y',NULL,NULL

FROM dba.TranSeg TS

INNER JOIN dba.InvoiceHeader IH

ON (IH.IataNum = TS.IataNum

AND IH.ClientCode = TS.ClientCode

AND IH.RecordKey = TS.RecordKey

AND IH.InvoiceDate = TS.InvoiceDate)

WHERE IH.IMPORTDT = @MaxImportDt

AND IH.IataNum = @IataNum

AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

AND TS.IataNum = @IataNum

AND TS.InvoiceDate between @FirstInvDate AND @LastInvDate

AND TS.SegmentCarrierCode IS NOT NULL

AND TS.ClassOfService IS NOT NULL

AND substring(TS.SegmentCarrierCode,1,3)+substring(TS.ClassOfService,1,1)+TS.SegInternationalInd

NOT IN (SELECT DISTINCT CarrierCode+ClassOfService+InternationalInd

FROM .DBA.ClassToCabin)

EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='ClassToCabin from Transeg',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

--------------------------------------------------------------------------

-------- Begin DELETE AND INSERT ComRmks -----

--------------------------------------------------------------------------

---- \*\*Note DO NOT ADD:

---- these JOINS:

---- AND IH.InvoiceDate = CR.InvoiceDate

---- AND IH.ClientCode = CR.ClientCode

---- or FILTER:

---- AND CR.InvoiceDate between @FirstInvDate AND @LastInvDate

---- in case InvoiceDates or ClientCodes changed with newly arrived data

---- \*\* Only match on Iatanum AND Recordkey to delete

SET @TransStart = getdate()

DELETE dba.ComRmks

FROM dba.ComRmks CR

INNER JOIN dba.InvoiceHeader IH

ON (IH.Iatanum = CR.Iatanum

AND IH.RecordKey = CR.RecordKey)

WHERE IH.IMPORTDT = @MaxImportDt

AND IH.IataNum = @IataNum

AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

AND CR.Iatanum = @IataNum

EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='Delete DBA.comrmks',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IATANUM=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

SET @TransStart = getdate()

INSERT INTO dba.ComRmks (RecordKey, IATANUM, SeqNum, ClientCode, InvoiceDate, IssueDate)

SELECT DISTINCT ID.RecordKey, ID.IATANUM, ID.SeqNum, ID.ClientCode, ID.InvoiceDate, ID.IssueDate

FROM dba.InvoiceDetail ID

INNER JOIN dba.InvoiceHeader IH

ON (IH.IataNum = ID.IataNum

AND IH.ClientCode = ID.ClientCode

AND IH.RecordKey = ID.RecordKey

AND IH.InvoiceDate = ID.InvoiceDate)

WHERE IH.IMPORTDT = @MaxImportDt

AND IH.IataNum = @IataNum

AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

AND ID.IataNum = @IataNum

AND ID.InvoiceDate between @FirstInvDate AND @LastInvDate

AND ID.RecordKey+ID.IataNum+CONVERT(VARCHAR,ID.SeqNum)

NOT IN (SELECT CR.RecordKey+CR.IataNum+CONVERT(VARCHAR,CR.SeqNum)

FROM DBA.ComRmks CR

WHERE CR.IataNum = @IataNum)

EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='Insert dba.ComRmks',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IATANUM=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

-----------------------------------------------------------------------

----------------------------------------------------------------------

---- ComRmks to 'Not Provided' prior to updating with agency data

----------------------------------------------------------------------

SET @TransStart = getdate()

UPDATE dba.comrmks

SET

Text1 = CASE WHEN Text1 is null then 'Not Provided' ELSE Text1 end,

Text2 = CASE WHEN Text2 is null then 'Not Provided' ELSE Text2 end,

Text3 = CASE WHEN Text3 is null then 'Not Provided' ELSE Text3 end,

Text4 = CASE WHEN Text4 is null then 'Not Provided' ELSE Text4 end,

Text5 = CASE WHEN Text5 is null then 'Not Provided' ELSE Text5 end,

Text6 = CASE WHEN Text6 is null then 'Not Provided' ELSE Text6 end,

Text7 = CASE WHEN Text7 is null then 'Not Provided' ELSE Text7 end,

Text8 = CASE WHEN Text8 is null then 'Not Provided' ELSE Text8 end,

Text9 = CASE WHEN Text9 is null then 'Not Provided' ELSE Text9 end,

Text10 = CASE WHEN Text10 is null then 'Not Provided' ELSE Text10 end,

Text11 = CASE WHEN Text11 is null then 'Not Provided' ELSE Text11 end,

Text12 = CASE WHEN Text12 is null then 'Not Provided' ELSE Text12 end,

Text13 = CASE WHEN Text13 is null then 'Not Provided' ELSE Text13 end,

text14 = CASE WHEN Text14 is null then 'Not Provided' ELSE Text14 end,

Text15 = CASE WHEN Text15 is null then 'Not Provided' ELSE Text15 end,

Text16 = CASE WHEN Text16 is null then 'Not Provided' ELSE Text16 end,

Text17 = CASE WHEN Text17 is null then 'Not Provided' ELSE Text17 end,

Text18 = CASE WHEN Text18 is null then 'Not Provided' ELSE Text18 end,

Text19 = CASE WHEN Text19 is null then 'Not Provided' ELSE Text19 end,

Text20 = CASE WHEN Text20 is null then 'Not Provided' ELSE Text20 end,

Text21 = CASE WHEN Text21 is null then 'Not Provided' ELSE Text21 end,

Text22 = CASE WHEN Text22 is null then 'Not Provided' ELSE Text22 end,

Text23 = CASE WHEN Text23 is null then 'Not Provided' ELSE Text23 end,

Text24 = CASE WHEN Text24 is null then 'Not Provided' ELSE Text24 end,

Text48 = CASE WHEN Text48 is null then 'Not Provided' ELSE Text48 end,

Text50 = CASE WHEN Text50 is null then 'Not Provided' ELSE Text50 end

FROM dba.ComRmks CR

INNER JOIN dba.InvoiceHeader IH

ON (IH.IataNum = CR.IataNum

AND IH.ClientCode = CR.ClientCode

AND IH.RecordKey = CR.RecordKey

AND IH.InvoiceDate = CR.InvoiceDate)

WHERE IH.IMPORTDT = @MaxImportDt

AND IH.IataNum = @IataNum

AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

AND CR.IataNum = @IataNum

AND CR.InvoiceDate between @FirstInvDate AND @LastInvDate

EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='SET text fields to Not Provided',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

----------------------------------------------------------------------

/\*

Text1 = Employee id

Text2 = POS country

Text3 = Cost Center

Text4 = Department

Text5 = Division/region

Text6 = Project code

Text7 = Trip purpose code

Text8 = Online indicator (usually the same as ID.OnlineBookingSystem)

Text9 =

Text10 =

Text11

Text12

Text13

Text14 = Highest cabin booked

Text15

Text16

Text17

Text18 = no hotel booked code

Text19

Text20

Text21

Text22

Text23

Text24

Text25

\*/

---------------------------------------------------------------------

--------------------------------------------------------

---- STANDARD Text2 with POS from InvoiceHeader OrigCountry

--------------------------------------------------------

SET @TransStart = getdate()

UPDATE CR

SET Text2 = CASE WHEN IH.OrigCountry IS NULL THEN 'Not Provided' ELSE IH.OrigCountry END

FROM dba.ComRmks CR

INNER JOIN dba.InvoiceHeader IH

ON (IH.IataNum = CR.IataNum

AND IH.ClientCode = CR.ClientCode

AND IH.RecordKey = CR.RecordKey

AND IH.InvoiceDate = CR.InvoiceDate)

WHERE IH.IMPORTDT = @MaxImportDt

AND IH.IataNum = @IataNum

AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

AND CR.IataNum = @IataNum

AND CR.InvoiceDate between @FirstInvDate AND @LastInvDate

EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='Text2\_IH POS\_Ctry',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

-----------------------------------------------------------------

----- End StANDard ComRmks Mappings

-----------------------------------------------------------------

-----------------------------------------------------------------

----- Begin client Specific ComRmks Mappings

-----------------------------------------------------------------

------------------------------------------------------------------------------------------------------------

---- Text1 for EmployeeID

---- Text50 for EmployeeID Raw Data as received without substrings other than 1,150 to fit in text field

---------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------

------ Text14 - STANDARD Highest Cabin

------ Only if Client Purchased - Verify with Implementation Consultant

-----------------------------------------------------------------

------ 1st) For First class

--SET @TransStart = getdate()

--UPDATE CR

--SET CR.Text14 = CTC.DomCabin

--FROM dba.ComRmks CR

--INNER JOIN dba.InvoiceHeader IH

-- ON (IH.Iatanum = CR.Iatanum

-- AND IH.Clientcode = CR.ClientCode

-- AND IH.RecordKey = CR.RecordKey

-- AND IH.InvoiceDate = CR.InvoiceDate)

--INNER JOIN dba.TranSeg TS

-- ON (CR.Iatanum = TS.Iatanum

-- AND CR.ClientCode = TS.ClientCode

-- AND CR.RecordKey = TS.RecordKey

-- AND CR.Seqnum = TS.SeqNum)

--INNER JOIN dba.classtocabin CTC

-- ON (TS.minsegmentcarriercode = CTC.carriercode

-- AND TS.minclassofservice = CTC.classofservice

-- AND TS.mininternationalind = CTC.InternationalInd)

--WHERE IH.IMPORTDT = @MaxImportDt

-- AND IH.iatanum = @IataNum

-- AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND CR.iatanum = @IataNum

-- AND CR.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND TS.IataNum = @IataNum

-- AND TS.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND CTC.DomCabin = 'First'

------AND CR.text14 = 'Not Provided'

--EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='Highest Cabin = FIRST',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

------ 2nd) For Business Class

--SET @TransStart = getdate()

--UPDATE CR

--SET CR.Text14 = CTC.DomCabin

--FROM dba.ComRmks CR

--INNER JOIN dba.InvoiceHeader IH

-- ON (IH.Iatanum = CR.Iatanum

-- AND IH.Clientcode = CR.ClientCode

-- AND IH.RecordKey = CR.RecordKey

-- AND IH.InvoiceDate = CR.InvoiceDate)

--INNER JOIN dba.TranSeg TS

-- ON (CR.Iatanum = TS.Iatanum

-- AND CR.ClientCode = TS.ClientCode

-- AND CR.RecordKey = TS.RecordKey

-- AND CR.Seqnum = TS.SeqNum)

--INNER JOIN dba.classtocabin CTC

-- ON (TS.minsegmentcarriercode = CTC.carriercode

-- AND TS.minclassofservice = CTC.classofservice

-- AND TS.mininternationalind = CTC.InternationalInd)

--WHERE IH.IMPORTDT = @MaxImportDt

-- AND IH.iatanum = @IataNum

-- AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND CR.iatanum = @IataNum

-- AND CR.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND TS.IataNum = @IataNum

-- AND TS.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND CTC.DomCabin = 'Business'

------AND CR.text14 = 'Not Provided'

--EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='Highest Cabin = BUSINESS',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR

------ 3rd) Then all others

--SET @TransStart = getdate()

--UPDATE CR

--SET CR.Text14 = CTC.DomCabin

--FROM dba.ComRmks CR

--INNER JOIN dba.InvoiceHeader IH

-- ON (IH.Iatanum = CR.Iatanum

-- AND IH.Clientcode = CR.ClientCode

-- AND IH.RecordKey = CR.RecordKey

-- AND IH.InvoiceDate = CR.InvoiceDate)

--INNER JOIN dba.TranSeg TS

-- ON (CR.Iatanum = TS.Iatanum

-- AND CR.ClientCode = TS.ClientCode

-- AND CR.RecordKey = TS.RecordKey

-- AND CR.Seqnum = TS.SeqNum)

--INNER JOIN dba.classtocabin CTC

-- ON (TS.minsegmentcarriercode = CTC.carriercode

-- AND TS.minclassofservice = CTC.classofservice

-- AND TS.mininternationalind = CTC.InternationalInd)

--WHERE IH.IMPORTDT = @MaxImportDt

-- AND IH.iatanum = @IataNum

-- AND IH.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND CR.iatanum = @IataNum

-- AND CR.InvoiceDate between @FirstInvDate AND @LastInvDate

-- AND TS.IataNum = @IataNum

-- AND TS.InvoiceDate between @FirstInvDate AND @LastInvDate

------AND CR.text14 = 'Not Provided'

--EXEC dbo.sp\_LogProcErrors @ProcedureName=@ProcName,@LogStart=@TransStart,@StepName='Highest Cabin = all others',@BeginDate=@BeginIssueDate,@EndDate=@EndIssueDate,@IataNum=@Iata,@RowCount=@@ROWCOUNT,@ERR=@@ERROR