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

CODE AUTHOR: Nathan Gambill

CODE TITLE: prc\_All\_DB\_Tables\_Columns\_Switch\_NOTNULL\_to\_NULL

DATE CREATED: December 5th, 2015

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

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

FACTS/ASSUMPTIONS:

o When Is\_nullable column schema property = zero (0) this means "NOT NULL" or "can’t contain NULLs", while a one (1) means "NULL" or "NULLs are allowed"; therefore you want to change zero's to 1's for most all columns

o You can NOT change Primary Key fields to "Null" ... they always are "NOT NULL"

o Probably good to leave all columns with a Column\_ID (or ordinal position) equal 1 alone if this is the tables acting primary key field --- in otherworld’s DONT attempt to change the main ID column of any table to "NULL"

o The following new code introduced in SQL Server 2012 is very helpful to find a columns schema details:

SELECT \* FROM sys.dm\_exec\_describe\_first\_result\_set (N'SELECT \* FROM CTSDM.dbo.vwFactTicket', null, 2)

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

--DROP PROCEDURE prc\_All\_DB\_Tables\_Columns\_Switch\_NOTNULL\_to\_NULL

--EXEC prc\_All\_DB\_Tables\_Columns\_Switch\_NOTNULL\_to\_NULL @DBName = 'TestDB'

CREATE PROCEDURE prc\_All\_DB\_Tables\_Columns\_Switch\_NOTNULL\_to\_NULL @DBName nvarchar(100)

AS

--Variable Declarations

DECLARE

@SQL\_Special\_INSERT\_INTO\_SELECT nvarchar(max)

--,@DBName nvarchar(100)

,@TableName nvarchar(100)

,@ColumnName nvarchar(100)

,@DataType nvarchar(100)

,@Curr\_Row\_Num int

,@Max\_Row\_Num int

,@Curr\_P\_ID int

,@Max\_P\_ID int

,@SQL\_ALTER\_STATEMENT nvarchar(500)

/\*\*\*\*\*\* INPUT THE NAME OF YOUR TARGET SYSTEM Database in here \*\*\*\*\*\*/

--SET @DBName = 'CTSDM'

/\*\*\*\*\*\* INPUT THE NAME OF YOUR TARGET SYSTEM Database above \*\*\*\*\*\*/

--CREATE a table to INSERT all the table names from the database in-focus

CREATE TABLE #temp\_TableList ( Row\_Num int, TABLE\_NAME nvarchar(100) )

INSERT INTO #temp\_TableList ( Row\_Num, TABLE\_NAME )

SELECT

Row\_Num = Row\_Number() OVER (ORDER BY Table\_Name)

,TABLE\_NAME

FROM INFORMATION\_SCHEMA.TABLES

WHERE

TABLE\_TYPE = 'BASE TABLE'

AND TABLE\_CATALOG='' + @DBName + ''

--SELECT \* FROM #temp\_TableList

--CREATE a TEMP table to hold the loop output for Loop #1

CREATE TABLE #temp\_Column\_Name\_and\_Type ( P\_ID int IDENTITY(1,1) PRIMARY KEY, ColumnOrdinalPosition int, DBName nvarchar(100), TableName nvarchar(100), ColumnName nvarchar(100), DataType nvarchar(200), Is\_Nullable int )

/\*\*\*\*\*\* FIRST LOOP \*\*\*\*\*\*\*/

--Set the loop control IDs to their initial settings

SET @Curr\_Row\_Num = 1

SET @Max\_Row\_Num = ( SELECT MAX(Row\_Num) FROM #temp\_TableList )

/\*\*\*\*\* FIRST LOOP BEGINS HERE \*\*\*\*\*/

WHILE @Curr\_Row\_Num <= @Max\_Row\_Num

--This loop gets all the table columns where the column is set to “NOT NULL” and puts the related info about that column into the Temp table you created above.

BEGIN

SET @TableName = ( SELECT Table\_Name FROM #temp\_TableList WHERE Row\_Num = @Curr\_Row\_Num )

SET @SQL\_Special\_INSERT\_INTO\_SELECT = N'(N''SELECT \* FROM ' + @DBName + '.dbo.' + @TableName + ' '''+', null, 2) '

--SELECT @SQL\_Special\_INSERT\_INTO\_SELECT

-- Use the “sys.dm\_exec\_describe\_first\_result\_set” SQL2012 feature below

SET @SQL\_Special\_INSERT\_INTO\_SELECT =

N'

INSERT INTO #temp\_Column\_Name\_and\_Type ( ColumnOrdinalPosition, DBName, TableName, ColumnName, DataType, Is\_Nullable )

SELECT

ColumnOrdinalPosition = Column\_Ordinal

,DBName = Source\_Database

,TableName = Source\_Table

,ColumnName = Name

,DataType = system\_type\_name

,Is\_Nullable

FROM sys.dm\_exec\_describe\_first\_result\_set ' + @SQL\_Special\_INSERT\_INTO\_SELECT + '

WHERE

Is\_Nullable = 0 --ONLY want the ones set to “NOT NULL” right now

AND Column\_Ordinal <> 1 --Do NOT want to chg. the main ID per table

AND Name <> ''RowStat'' --this is a special row I do NOT want to get

; '

--SELECT @SQL\_Special\_INSERT\_INTO\_SELECT

EXEC sp\_executesql @SQL\_Special\_INSERT\_INTO\_SELECT

--Reset the Loop

SET @Curr\_Row\_Num = @Curr\_Row\_Num + 1

END

--SELECT \* FROM #temp\_Column\_Name\_and\_Type

/\*\*\*\*\* FIRST LOOP ENDS HERE \*\*\*\*\*/

/\*\*\*\*\*\* SECOND LOOP \*\*\*\*\*\*\*/

/\*\*\*\*\* SECOND LOOP BEGINS HERE \*\*\*\*\*/

SET @Curr\_P\_ID = 1

SET @Max\_P\_ID = ( SELECT MAX(P\_ID) FROM #temp\_Column\_Name\_and\_Type )

WHILE @Curr\_P\_ID <= @Max\_P\_ID

--This loop uses the above collected temp table data to BUILD and EXEC every needed ALTER statement to set all the “NOT NULLs” to “NULL”

BEGIN

SET @TableName = ( SELECT TableName FROM #temp\_Column\_Name\_and\_Type WHERE P\_ID = @Curr\_P\_ID )

SET @ColumnName = ( SELECT ColumnName FROM #temp\_Column\_Name\_and\_Type WHERE P\_ID = @Curr\_P\_ID )

SET @DataType = ( SELECT DataType FROM #temp\_Column\_Name\_and\_Type WHERE P\_ID = @Curr\_P\_ID )

SET @SQL\_ALTER\_STATEMENT = 'ALTER TABLE [' + @DBName + '].[dbo].[' + @TableName + '] ALTER COLUMN ' + @ColumnName + ' ' + @DataType + ' NULL -- ' + CONVERT(nvarchar,@Curr\_P\_ID)

--PRINT @SQL\_ALTER\_STATEMENT

EXECUTE sp\_executesql @SQL\_ALTER\_STATEMENT

--Reset the Loop

SET @Curr\_P\_ID = @Curr\_P\_ID + 1

END

/\*\*\*\*\* SECOND LOOP ENDS HERE \*\*\*\*\*/

/\*\*\*\*\*\*\* DROP THE ABOVE TEMP TABLES \*\*\*\*\*\*\*/

DROP TABLE #temp\_TableList

DROP TABLE #temp\_Column\_Name\_and\_Type

/\*\*\*\*\* END OF PROC \*\*\*\*\*\*/

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

--Used to get a quick look at what the basic "sys.dm\_exec\_describe\_first\_result\_set" will return on a given table

SELECT \*

FROM sys.dm\_exec\_describe\_first\_result\_set

(N'SELECT \* FROM CTSDM.dbo.vwFactTicket', null, 2)

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

**BELOW IS TEST CODE USED IN THE VIDEO AND IS NOT NEEDED AS PART OF THE ABOVE PROCEDURE**

--1. Make sure our Test tables are all removed

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

DROP TABLE TestDB.dbo.SOURCE\_tblFactTicket

DROP TABLE TestDB.dbo.SOURCE\_tblDimCustomer

DROP TABLE TestDB.dbo.SOURCE\_tblDimEmployee

DROP TABLE TestDB.dbo.TARGET\_tblFactTicket

DROP TABLE TestDB.dbo.TARGET\_tblDimCustomer

DROP TABLE TestDB.dbo.TARGET\_tblDimEmployee

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

--2. Create Source tables. Note: these SOURCE tables would already be in-place in a real world environment.

CREATE TABLE TestDB.dbo.SOURCE\_tblFactTicket (

TicketDerivedID int IDENTITY(1,1) PRIMARY KEY

,TicketID int NULL

,CustomerDerivedID int NULL

,CurrentEmployeeDerivedID int NULL

,CallDurationMins int NULL

,CaseNotes nvarchar(255) NULL

,CaseOpneCloseStatus char(10) NULL

,CustomerContactMethod varchar(35) NULL

,CaseOpenDateTime datetime NULL

,CaseClosedDateTime datetime NULL

)

CREATE TABLE TestDB.dbo.SOURCE\_tblDimCustomer (

CustomerDerivedID int IDENTITY(1,1) PRIMARY KEY

,TicketDerivedID int NULL

,CustomerLName nvarchar(50) NULL

,CustomerFName nvarchar(50) NULL

,CustomerMI nvarchar(1) NULL

,CustomerEmail nvarchar(100) NULL

,CustomerSpecialNotes nvarchar(255) NULL

,CustomerAcctCreationDateTime datetime NULL

,CustomerActive smallint NULL

)

CREATE TABLE TestDB.dbo.SOURCE\_tblDimEmployee (

EmployeeDerivedID int IDENTITY(1,1) PRIMARY KEY

,EmployeeLName nvarchar(50) NULL

,EmployeeFName nvarchar(50) NULL

,EmployeeMI nvarchar(1) NULL

,EmployeeEmail nvarchar(100) NULL

,EmployeeSpecialNotes nvarchar(255) NULL

,EmployeeAcctCreationDateTime datetime NULL

,EmployeeActive smallint NULL

,EmployeeRoleTitle nvarchar(100) NULL

)

--3. Insert some made up test data into Source tables. Note: this SOURCE table data would already be in-place in a real world environment.

INSERT INTO TestDB.dbo.SOURCE\_tblFactTicket VALUES(1111111111,1,1,24.45,'Call back soon','Closed','Phone',42143.3487539352,42143.3487539352)

INSERT INTO TestDB.dbo.SOURCE\_tblFactTicket VALUES(1111111112,2,2,12.9,'They liked our ultimate software','Closed','Phone',42299.3904166667,42299.3904166667)

INSERT INTO TestDB.dbo.SOURCE\_tblFactTicket VALUES(1111111113,3,3,3.87,'send file as soon as scoping completes','Closed','Web Portal',42040.4320833333,42040.4320833333)

INSERT INTO TestDB.dbo.SOURCE\_tblFactTicket VALUES(1111111114,4,4,1.09,'Bad attitude-- need recovery','Open','Web Portal',42333.47375,NULL)

INSERT INTO TestDB.dbo.SOURCE\_tblFactTicket VALUES(1111111115,5,5,23.65,'previous Eng. was very helpful','Closed','Email',42339.5154166667,42339.5154166667)

INSERT INTO TestDB.dbo.SOURCE\_tblFactTicket VALUES(1111111116,5,5,NULL,NULL,NULL,NULL,NULL,NULL)

INSERT INTO TestDB.dbo.SOURCE\_tblDimCustomer VALUES(1,'Jones','Eddie','G',' EddieGJones999@gmail.com','Happy',41914.5316512732,1)

INSERT INTO TestDB.dbo.SOURCE\_tblDimCustomer VALUES(2,'Stevens','Carl','C',' CarlCStevens@hotmail.com','Moderate',41914.5733179398,1)

INSERT INTO TestDB.dbo.SOURCE\_tblDimCustomer VALUES(3,'Snovel','Susan','E',' SusanESnovel@yahoo.com','Nice',41914.6149846065,1)

INSERT INTO TestDB.dbo.SOURCE\_tblDimCustomer VALUES(4,'Baker','Melinda','W',' MelindaWBaker@andersonsinc.com','Pleasing',41914.6566512732,1)

INSERT INTO TestDB.dbo.SOURCE\_tblDimCustomer VALUES(5,'Carter','Bob','O',' BobOCarter@usgovmildat.com','Very Cooperative',41914.6983179398,1)

INSERT INTO TestDB.dbo.SOURCE\_tblDimCustomer VALUES(6,NULL,NULL,NULL,NULL,NULL,41914.6983179398,NULL)

INSERT INTO TestDB.dbo.SOURCE\_tblDimEmployee VALUES('Fitzgerald','Mike','E','MikeEFitzgerald@HarvestorSupt.com','No allergies',41866.5316550926,1,'Support Engineer')

INSERT INTO TestDB.dbo.SOURCE\_tblDimEmployee VALUES('Cooper','Steve','R','SteveRCooper@HarvestorSupt.com','No allergies',41867.5316550926,1,'Sr. Support Engineer')

INSERT INTO TestDB.dbo.SOURCE\_tblDimEmployee VALUES('Howel','Cary','G','CaryGHowel@HarvestorSupt.com','No allergies',41868.5316550926,1,'Support Engineer')

INSERT INTO TestDB.dbo.SOURCE\_tblDimEmployee VALUES('Otoberger','Elizabeth','C','ElizabethCOtoberger@HarvestorSupt.com','Allergic to nuts',41869.5316550926,1,'Technical Account Manager')

INSERT INTO TestDB.dbo.SOURCE\_tblDimEmployee VALUES('Calenger','Fred','A','FredACalenger@HarvestorSupt.com','No allergies',41870.5316550926,1,'Support Engineer')

INSERT INTO TestDB.dbo.SOURCE\_tblDimEmployee VALUES('Calenger',NULL,NULL,NULL,NULL,41870.5316550926,NULL,NULL)

--4. See the data has been put into the SOURCE tables.

SELECT \* FROM TestDB.dbo.SOURCE\_tblFactTicket

SELECT \* FROM TestDB.dbo.SOURCE\_tblDimCustomer

SELECT \* FROM TestDB.dbo.SOURCE\_tblDimEmployee

--5. Create TARGET tables.

--We often use something like this ...

--SELECT TOP 1000 \* INTO TestDB.dbo.TARGET\_tblFactTicket FROM TestDB.dbo.SOURCE\_tblFactTicket

--to initially build our TARGET tables, then TRUNCATE, then fill using a daily SSIS Pkg

--problem occurs when the TARGET table columns upon inital creation assumes many "NOT NULL" settings when they are in fact set to "NULL" on the SOURCE side.

CREATE TABLE TestDB.dbo.TARGET\_tblFactTicket (

TicketDerivedID int IDENTITY(1,1) PRIMARY KEY

,TicketID int NOT NULL

,CustomerDerivedID int NOT NULL

,CurrentEmployeeDerivedID int NOT NULL

,CallDurationMins int NOT NULL

,CaseNotes nvarchar(255) NOT NULL

,CaseOpneCloseStatus char(10) NOT NULL

,CustomerContactMethod varchar(35) NOT NULL

,CaseOpenDateTime datetime NOT NULL

,CaseClosedDateTime datetime NOT NULL

)

CREATE TABLE TestDB.dbo.TARGET\_tblDimCustomer (

CustomerDerivedID int IDENTITY(1,1) PRIMARY KEY

,TicketDerivedID int NOT NULL

,CustomerLName nvarchar(50) NOT NULL

,CustomerFName nvarchar(50) NOT NULL

,CustomerMI nvarchar(1) NOT NULL

,CustomerEmail nvarchar(100) NOT NULL

,CustomerSpecialNotes nvarchar(255) NOT NULL

,CustomerAcctCreationDateTime datetime NOT NULL

,CustomerActive smallint NOT NULL

)

CREATE TABLE TestDB.dbo.TARGET\_tblDimEmployee (

EmployeeDerivedID int IDENTITY(1,1) PRIMARY KEY

,EmployeeLName nvarchar(50) NOT NULL

,EmployeeFName nvarchar(50) NOT NULL

,EmployeeMI nvarchar(1) NOT NULL

,EmployeeEmail nvarchar(100) NOT NULL

,EmployeeSpecialNotes nvarchar(255) NOT NULL

,EmployeeAcctCreationDateTime datetime NOT NULL

,EmployeeActive smallint NOT NULL

,EmployeeRoleTitle nvarchar(100) NOT NULL

)

--6. Insert some made up test data into TARGET tables.

INSERT INTO TestDB.dbo.TARGET\_tblFactTicket VALUES(1111111111,1,1,24.45,'Call back soon','Closed','Phone',42143.3487539352,42143.3487539352)

INSERT INTO TestDB.dbo.TARGET\_tblFactTicket VALUES(1111111112,2,2,12.9,'They liked our ultimate software','Closed','Phone',42299.3904166667,42299.3904166667)

INSERT INTO TestDB.dbo.TARGET\_tblFactTicket VALUES(1111111113,3,3,3.87,'send file as soon as scoping completes','Closed','Web Portal',42040.4320833333,42040.4320833333)

INSERT INTO TestDB.dbo.TARGET\_tblFactTicket VALUES(1111111114,4,4,1.09,'Bad attitude-- need recovery','Open','Web Portal',42333.47375,NULL)

INSERT INTO TestDB.dbo.TARGET\_tblFactTicket VALUES(1111111115,5,5,23.65,'previous Eng. was very helpful','Closed','Email',42339.5154166667,42339.5154166667)

INSERT INTO TestDB.dbo.TARGET\_tblFactTicket VALUES(1111111116,5,5,NULL,NULL,NULL,NULL,NULL,NULL)

INSERT INTO TestDB.dbo.TARGET\_tblDimCustomer VALUES(1,'Jones','Eddie','G',' EddieGJones999@gmail.com','Happy',41914.5316512732,1)

INSERT INTO TestDB.dbo.TARGET\_tblDimCustomer VALUES(2,'Stevens','Carl','C',' CarlCStevens@hotmail.com','Moderate',41914.5733179398,1)

INSERT INTO TestDB.dbo.TARGET\_tblDimCustomer VALUES(3,'Snovel','Susan','E',' SusanESnovel@yahoo.com','Nice',41914.6149846065,1)

INSERT INTO TestDB.dbo.TARGET\_tblDimCustomer VALUES(4,'Baker','Melinda','W',' MelindaWBaker@andersonsinc.com','Pleasing',41914.6566512732,1)

INSERT INTO TestDB.dbo.TARGET\_tblDimCustomer VALUES(5,'Carter','Bob','O',' BobOCarter@usgovmildat.com','Very Cooperative',41914.6983179398,1)

INSERT INTO TestDB.dbo.TARGET\_tblDimCustomer VALUES(6,NULL,NULL,NULL,NULL,NULL,41914.6983179398,NULL)

INSERT INTO TestDB.dbo.TARGET\_tblDimEmployee VALUES('Fitzgerald','Mike','E','MikeEFitzgerald@HarvestorSupt.com','No allergies',41866.5316550926,1,'Support Engineer')

INSERT INTO TestDB.dbo.TARGET\_tblDimEmployee VALUES('Cooper','Steve','R','SteveRCooper@HarvestorSupt.com','No allergies',41867.5316550926,1,'Sr. Support Engineer')

INSERT INTO TestDB.dbo.TARGET\_tblDimEmployee VALUES('Howel','Cary','G','CaryGHowel@HarvestorSupt.com','No allergies',41868.5316550926,1,'Support Engineer')

INSERT INTO TestDB.dbo.TARGET\_tblDimEmployee VALUES('Otoberger','Elizabeth','C','ElizabethCOtoberger@HarvestorSupt.com','Allergic to nuts',41869.5316550926,1,'Technical Account Manager')

INSERT INTO TestDB.dbo.TARGET\_tblDimEmployee VALUES('Calenger','Fred','A','FredACalenger@HarvestorSupt.com','No allergies',41870.5316550926,1,'Support Engineer')

INSERT INTO TestDB.dbo.TARGET\_tblDimEmployee VALUES('Calenger',NULL,NULL,NULL,NULL,41870.5316550926,NULL,NULL)

--7. One solution is to just loop through all the TARGET tables and columns and set them all to "NULL" (or so they allow Null entries).