create PROCEDURE [dbo].[StaticLoad\_**PAStatus**]

AS

BEGIN

set nocount on;

with CTE\_Updates ( Code, Name, Description)

as

(

Select 'N/A' as Code, 'N/A' as Name, 'N/A' as Description

union all Select 'Not Reques' as Code, 'Not Requested' as Name, 'Not Requested' as Description

union all Select 'Requested' as Code, 'Requested' as Name, 'Requested' as Description

)

-- Change 03 Change Table name

Merge **PAStatus** as Target

Using CTE\_Updates as Source

On Target.CODE = Source.CODE

WHEN MATCHED THEN

Update

Set

Target.Code = Source.Code,

Target.Name = Source.Name,

Target.Description = Source.Description

WHEN NOT MATCHED THEN

Insert (

Code,

Name,

Description

)

Values (

Source.Code,

Source.Name,

Source.Description

)

;

END

= = = = = = = = = = =

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

/\*

CREATE PROCEDURE  [dbo].[sproc\_Generate\_StaticLoadTable]

(

    @sparm\_Message        varchar(2000) output

)

AS

BEGIN

    set nocount on;

    declare @intReturnCode             int = 0;

    declare @strMessage\_Successful varchar(1) = ' ';

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

    -- establish stored procedure name

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

    declare @strThisStoredProcedureName varchar(128) = object\_name(@@procid);

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

    -- establish transaction name if needed

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

    declare @strTransactionName            varchar(32)  = @strThisStoredProcedureName;

    declare @strDummy char(1);

    BEGIN TRY

         -------

         -- establish cursor variables for tracking

         -------

         declare @cursorStatus\_Open         smallint =  1;

         declare @cursorStatus\_Closed       smallint = -1;

         declare @cursorStatus\_DoesNotExist smallint = -3;

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

         -- Get all the ticket data from bulk load insert which is occurring.

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

         declare @intCursor\_TicketId    int          = 0;

         declare @intReturnCodeExpunged int          = 0;

-- change 05 - make sure datatypes match EXACTLY - especially Code

         declare @Code                  nvarchar(10);

         declare @Name                  nvarchar(50);

         declare @Description           nvarchar(max);

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

         declare @strStringToSendBack   varchar(max) = ' ';

-- change 06 - Change table name in cursor

         DECLARE Cursor\_Data CURSOR local FAST\_FORWARD

             FOR

                 select Code, Name, Description from **PAStatus** (nolock);

         OPEN Cursor\_Data;

         FETCH NEXT   FROM Cursor\_Data INTO @Code, @Name, @Description;

         declare @intCounter int = 0;

         WHILE @@FETCH\_STATUS = 0

             BEGIN

                 if  (@intCounter = 0)

                      begin

                          set @strDummy = ' ';

                      end

                 else

                      begin

                          set @strStringToSendBack += ' union all ';

                      end

                 set @intCounter += 1;

                 set @strStringToSendBack +=

                          ' Select ' + '''' + @Code + '''' + ' as Code, ' +

                                        '''' + @Name +  ''''  + ' as Name, ' +

                                       '''' + @Description + '''' + ' as Description ';

                 -------

                 -- Get next row from cursor

                 -------

                 FETCH NEXT   FROM Cursor\_Data INTO @code, @Name, @Description;

             END

         CLOSE        Cursor\_Data;

         DEALLOCATE   Cursor\_Data;

         select @strStringToSendBack as UpdateString;

         -- send message to caller for success

         set @sparm\_Message = @strMessage\_Successful;

    END TRY

    BEGIN CATCH

         DECLARE @ErrorMessage VARCHAR(MAX) = ERROR\_MESSAGE();

         DECLARE @ErrorSeverity    INT     = ERROR\_SEVERITY();

         DECLARE @ErrorState       INT      = ERROR\_STATE();

         set @intReturnCode    = ERROR\_NUMBER();

         set @sparm\_Message    = @ErrorMessage; -- send error message to caller

         -- signal to SSIS package that the sproc failed

         RAISERROR    (

                          @ErrorMessage,

                          @ErrorSeverity,

                          @ErrorState

                      ) with seterror;

    END CATCH

    return isnull(@intReturnCode, 0);

END

The result generates

Select 'N/A' as Code, 'N/A' as Name, 'N/A' as Description

union all Select 'Not Reques' as Code, 'Not Requested' as Name, 'Not Requested' as Description

union all Select 'Requested' as Code, 'Requested' as Name, 'Requested' as Description

union all Select 'Approved' as Code, 'Approved' as Name, 'Approved' as Description

union all Select 'Denied' as Code, 'Denied' as Name, 'Denied' as Description