USE [ATLAS]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[EN\_UpdateShorts] Script Date: 11/1/2019 5:35:24 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER procedure [dbo].[EN\_UpdateShorts] as

--DROP TABLE IF EXISTS ATLAS.dbo.A\_ShortContractTest ;

--select \* into ATLAS.dbo.A\_ShortContractTest from ATLAS.dbo.A\_ShortContractTable

--select \* from ATLAS.dbo.A\_ShortContractTest

declare @XCount int

set @XCount = (select count(\*) from ATLAS.dbo.A\_ShortContractStaging)

--print @XCount

if (@XCount > 1000)

BEGIN

MERGE ATLAS.dbo.A\_ShortContractTable WITH (HOLDLOCK) As TARGET

USING (select \* from ATLAS.dbo.A\_ShortContractStaging where [Store#] is not null) AS SOURCE

ON (TARGET.[Store#] = SOURCE.[Store#] and TARGET.[Contract #] = SOURCE.[Contract #] and TARGET.[Expiration Date] = SOURCE.[Expiration Date] )

--When records are matched, update the records if there is any change

WHEN MATCHED AND

TARGET.Supplier <> SOURCE.Supplier

OR TARGET.[Owned By] <> SOURCE.[Owned By]

OR TARGET.[Contract #] <> SOURCE.[Contract #]

OR TARGET.[Contract Type] <> SOURCE.[Contract Type]

OR TARGET.[Start Date] <> SOURCE.[Start Date]

OR TARGET.[Yearly Invoice Amount] <> SOURCE.[Yearly Invoice Amount]

OR TARGET.[Expiration Date] <> SOURCE.[Expiration Date]

OR TARGET.[Store Name] <> SOURCE.[Store Name]

OR TARGET.[Region] <> SOURCE.[Region]

OR TARGET.[PAYTYPE] <> SOURCE.[PAYTYPE]

OR TARGET.[XXX] <> SOURCE.[XXX]

OR TARGET.[Manufacturer] <> SOURCE.[Manufacturer]

OR TARGET.[AgreementDoc] <> SOURCE.[AgreementDoc]

THEN UPDATE SET TARGET.Supplier = SOURCE.Supplier,

TARGET.[Owned By] = SOURCE.[Owned By],

TARGET.[Contract #] = SOURCE.[Contract #],

TARGET.[Contract Type] = SOURCE.[Contract Type],

TARGET.[Start Date] = SOURCE.[Start Date],

TARGET.[Yearly Invoice Amount] = SOURCE.[Yearly Invoice Amount],

TARGET.[Expiration Date] = SOURCE.[Expiration Date],

TARGET.[Store Name] = SOURCE.[Store Name],

TARGET.[Region] = SOURCE.[Region],

TARGET.[PAYTYPE] = SOURCE.[PAYTYPE],

TARGET.[XXX] = SOURCE.[XXX],

TARGET.[Manufacturer] = SOURCE.[Manufacturer],

TARGET.[AgreementDoc] = SOURCE.[AgreementDoc]

--When no records are matched, insert the incoming records from source table to target table

WHEN NOT MATCHED BY TARGET

THEN INSERT ([Store#], Supplier, [Owned By], [Contract #], [Contract Type], [Start Date],

[Yearly Invoice Amount], [Expiration Date], [Store Name],

[Region], [PAYTYPE], [XXX], [Manufacturer], [AgreementDoc] )

VALUES

(SOURCE.[Store#], SOURCE.Supplier, SOURCE.[Owned By], SOURCE.[Contract #], SOURCE.[Contract Type], SOURCE.[Start Date],

SOURCE.[Yearly Invoice Amount], SOURCE.[Expiration Date], SOURCE.[Store Name],

SOURCE.[Region], SOURCE.[PAYTYPE], SOURCE.[XXX], SOURCE.[Manufacturer], SOURCE.[AgreementDoc] )

--When there is a row that exists in target and same record does not exist in source then delete this record target

WHEN NOT MATCHED BY SOURCE

THEN DELETE

--$action specifies a column of type nvarchar(10) in the OUTPUT clause that returns

--one of three values for each row: 'INSERT', 'UPDATE', or 'DELETE' according to the action that was performed on that row

--OUTPUT $action,

--DELETED.[Store#] AS TargetStore,

--INSERTED.[Store#] AS SourceStore;

--SELECT @@ROWCOUNT;

;

END

GO