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

 COPYRIGHTS ssawalha@uba.com.jo

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

CREATE PROCEDURE Delete\_Folder

(

 @FolderName varchar(100)

 ,@WorkTest char(1) = 'w' -- use 'w' to work and 't' to print

)

AS

/\*-----------------------------------------------------------------------------------------

 Author : Sahem Sawalha

 Date: 20/12/2019

 Description: It drop all the objects in a schema and then the schema itself

-------------------------------------------------------------------------------------------\*/

BEGIN

declare @SQL varchar(4000)

declare @msg varchar(500)

IF OBJECT\_ID('tempdb..#dropcode') IS NOT NULL DROP TABLE #dropcode

CREATE TABLE #dropcode

(

 ID int identity(1,1)

 ,SQLstatement varchar(1000)

 )

-- removes all the foreign keys that reference a PK in the target schema

 SELECT @SQL =

 'select

 '' ALTER TABLE ''+SCHEMA\_NAME(fk.schema\_id)+''.''+OBJECT\_NAME(fk.parent\_object\_id)+'' DROP CONSTRAINT ''+ fk.name

 FROM sys.foreign\_keys fk

 join sys.tables t on t.object\_id = fk.referenced\_object\_id

 where t.schema\_id = schema\_id(''' + @FolderName+''')

 and fk.schema\_id <> t.schema\_id

 order by fk.name desc'

 IF @WorkTest = 't' PRINT (@SQL )

 INSERT INTO #dropcode

 EXEC (@SQL)

 -- drop all default constraints, check constraints and Foreign Keys

 SELECT @SQL =

 'SELECT

 '' ALTER TABLE ''+schema\_name(t.schema\_id)+''.''+OBJECT\_NAME(fk.parent\_object\_id)+'' DROP CONSTRAINT ''+ fk.[Name]

 FROM sys.objects fk

 join sys.tables t on t.object\_id = fk.parent\_object\_id

 where t.schema\_id = schema\_id(''' + @FolderName+''')

 and fk.type IN (''D'', ''C'', ''F'')'

 IF @WorkTest = 't' PRINT (@SQL )

 INSERT INTO #dropcode

 EXEC (@SQL)

 -- drop all other objects in order

 SELECT @SQL =

 'SELECT

 CASE WHEN SO.type=''PK'' THEN '' ALTER TABLE ''+SCHEMA\_NAME(SO.schema\_id)+''.''+OBJECT\_NAME(SO.parent\_object\_id)+'' DROP CONSTRAINT ''+ SO.name

 WHEN SO.type=''U'' THEN '' DROP TABLE ''+SCHEMA\_NAME(SO.schema\_id)+''.''+ SO.[Name]

 WHEN SO.type=''V'' THEN '' DROP VIEW ''+SCHEMA\_NAME(SO.schema\_id)+''.''+ SO.[Name]

 WHEN SO.type=''P'' THEN '' DROP PROCEDURE ''+SCHEMA\_NAME(SO.schema\_id)+''.''+ SO.[Name]

 WHEN SO.type=''TR'' THEN '' DROP TRIGGER ''+SCHEMA\_NAME(SO.schema\_id)+''.''+ SO.[Name]

 WHEN SO.type IN (''FN'', ''TF'',''IF'',''FS'',''FT'') THEN '' DROP FUNCTION ''+SCHEMA\_NAME(SO.schema\_id)+''.''+ SO.[Name]

 WHEN SO.type=''SO'' THEN '' DROP SEQUENCE ''+SCHEMA\_NAME(SO.schema\_id)+''.''+ SO.[Name]

 END

FROM sys.objects SO

WHERE SO.schema\_id = schema\_id('''+ @FolderName +''')

 AND SO.type IN (''PK'', ''FN'', ''TF'', ''TR'', ''V'', ''U'', ''P'',''SO'')

ORDER BY CASE WHEN type = ''PK'' THEN 1

 WHEN type in (''FN'', ''TF'', ''P'',''IF'',''FS'',''FT'') THEN 2

 WHEN type = ''TR'' THEN 3

 WHEN type = ''V'' THEN 4

 WHEN type = ''U'' THEN 5

 WHEN type = ''SO'' THEN 6

 ELSE 7

 END'

IF @WorkTest = 't' PRINT (@SQL )

INSERT INTO #dropcode

EXEC (@SQL)

DECLARE @ID int, @statement varchar(1000)

DECLARE statement\_cursor CURSOR

FOR SELECT SQLstatement

 FROM #dropcode

 ORDER BY ID ASC

 OPEN statement\_cursor

 FETCH statement\_cursor INTO @statement

 WHILE (@@FETCH\_STATUS = 0)

 BEGIN

 IF @WorkTest = 't' PRINT (@statement)

 ELSE

 BEGIN

 PRINT (@statement)

 EXEC(@statement)

 END

 FETCH statement\_cursor INTO @statement

END

CLOSE statement\_cursor

DEALLOCATE statement\_cursor

IF @WorkTest = 't' PRINT ('DROP SCHEMA '+@FolderName)

ELSE

 BEGIN

 PRINT ('DROP SCHEMA '+@FolderName)

 EXEC ('DROP SCHEMA '+@FolderName)

 END

PRINT '------- ALL - DONE -------'

END