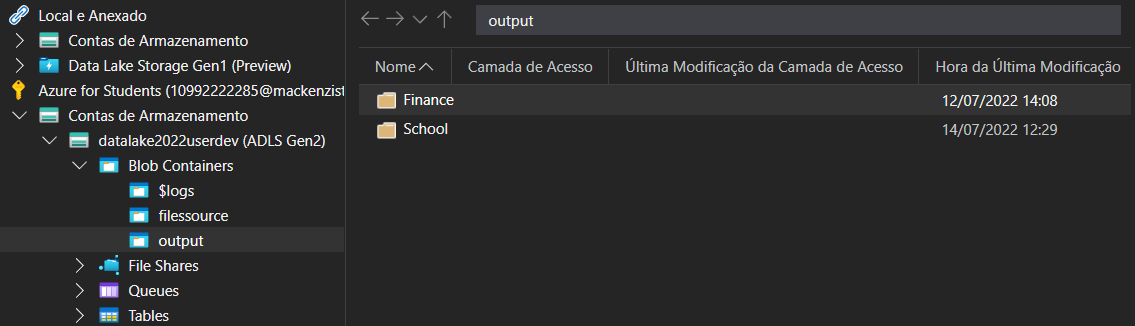
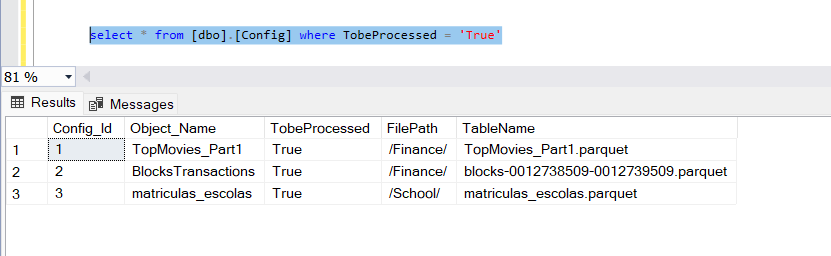
DATA FACTORY – BULK ISERT – LOOK UP + FOR EACH COM TABELA CONTROLE SQL

(FROM AZURE BLOB STORAGE / LAKE - MULTIPLE .PARQUET TO SQL TABLE)



**1º** Criar a Tabela Config que será a tabela aonde o pipeline irá ler os metadados e fazer match com os arquivos parquet no blob storage. Essa Tabela também serve para controle de carga ex: ( posso configurar quais tabelas quero efetuar a carga , passando parametros de schema, nome de tabela, folder e flags):



CREATE TABLE [dbo].[Config](

[Config\_Id] [int] IDENTITY(1,1) NOT NULL,

[Object\_Name] [varchar](250) NULL,

[TobeProcessed] [varchar](20) NULL,

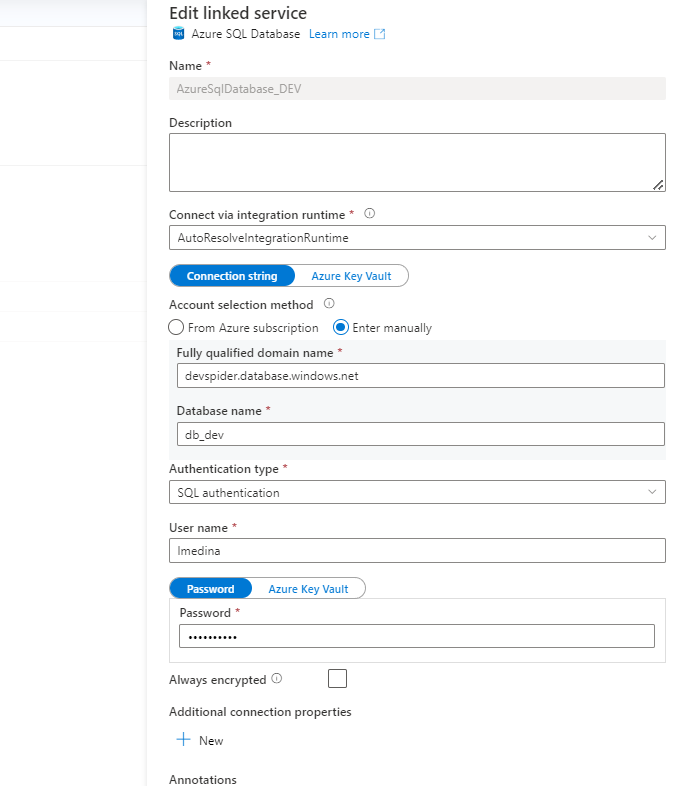
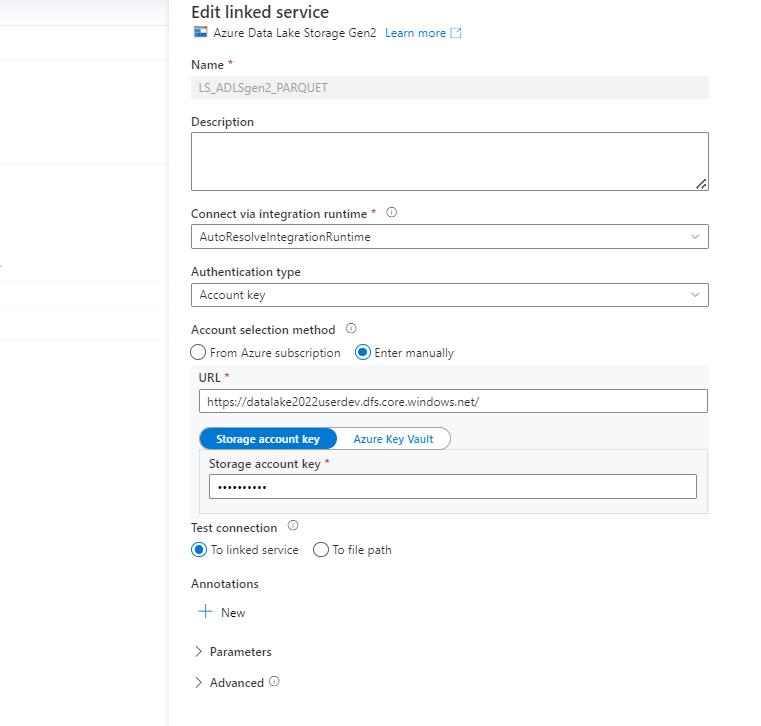
[FilePath] [varchar](100) NULL,

[TableName] [varchar](100) NULL

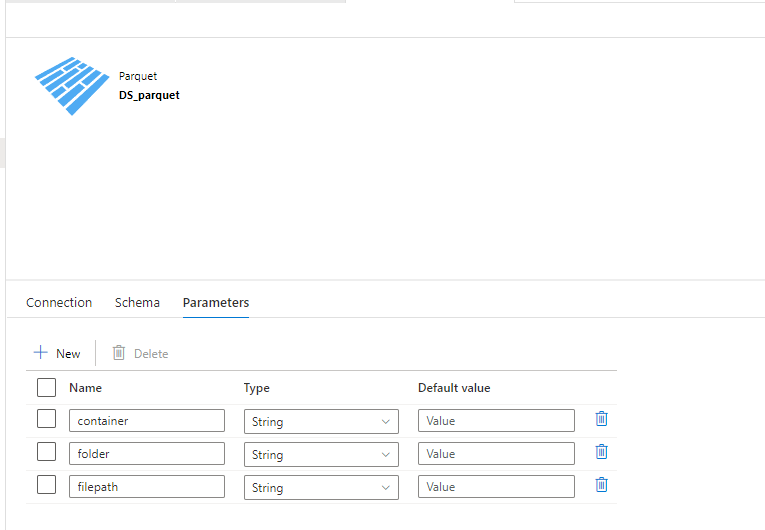
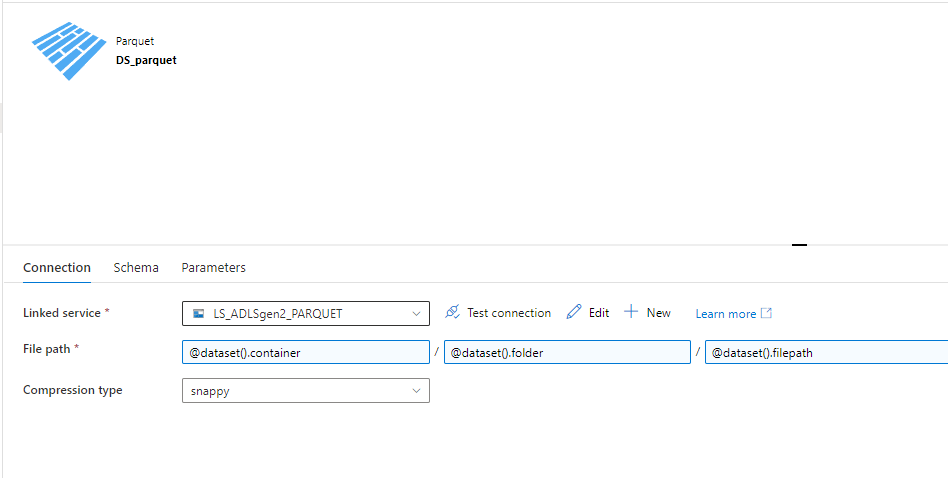
) ON [PRIMARY]

GO

**2º** Criar linked server para azure blob storage/lake e sql server:

**3°** Criar data set para azure blob/lake e criar as variaveis dinâmicas:

**Parameters:**

container

folder

filepath

**File Path:**

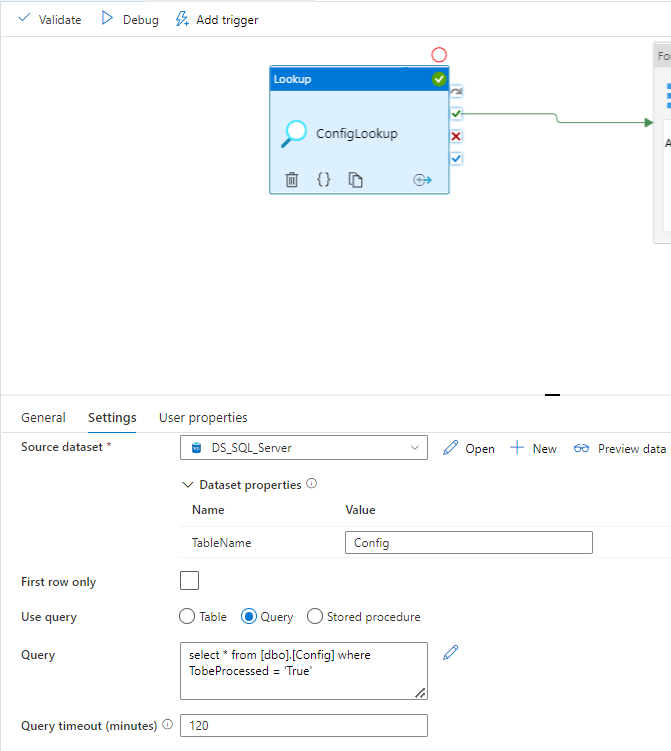
@dataset().container

@dataset().folder

@dataset().filepath

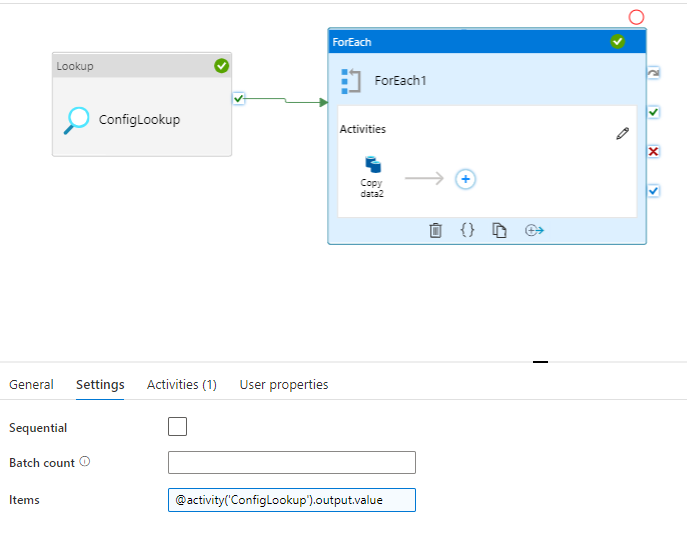
**4°** Criar Pipeline :

- LookUp ( Aonde o For Each irá ler a tabela Config ára dar match com os arquivos parquet no blobstorage e então fazer o array/looping de insert dinâmico ) :



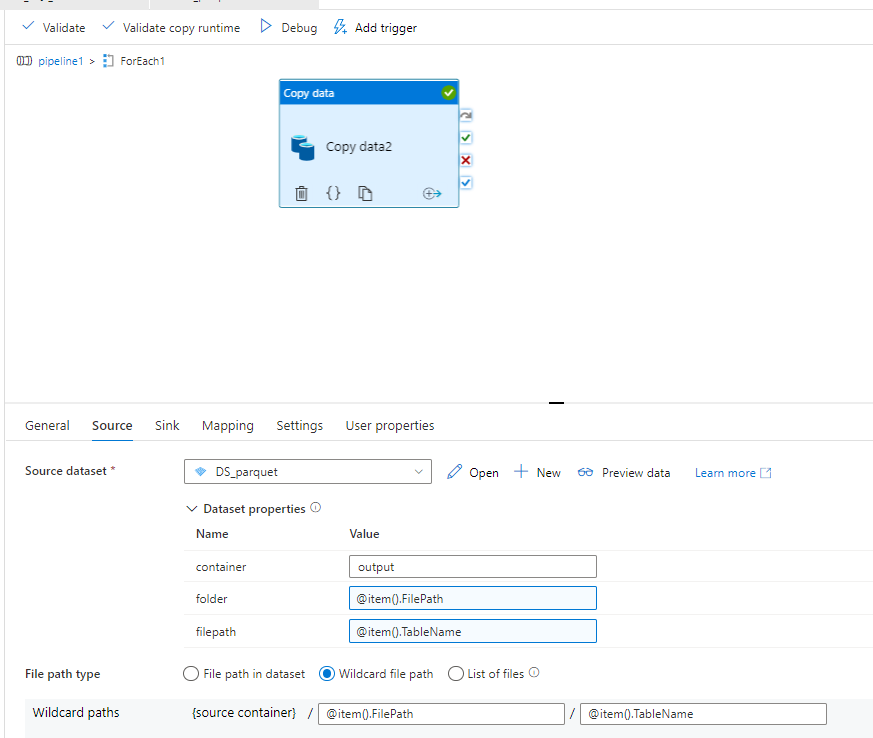
select \* from [dbo].[Config] where TobeProcessed = 'True'

Criar ForEach :

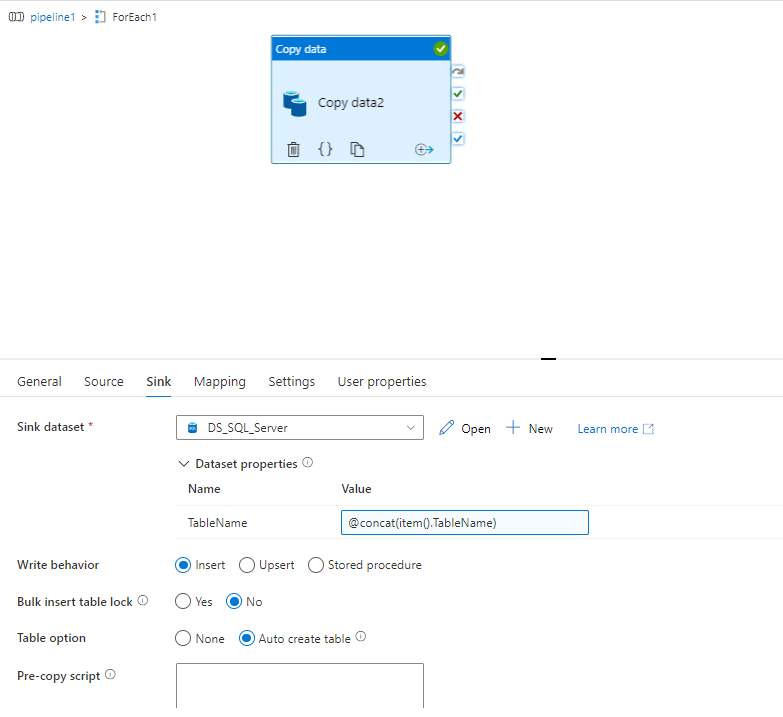


**Itens**: @**activity**('ConfigLookup').output.value

Criar Activity Copy (Source) :



Activity Copy (Sink:) :



TableName: @**concat**(**item**().TableName)

Done.

