Below is the SQL code to reproduce this updated pipeline\_parameter table:

SET ansi\_nulls ON  
  
go  
  
SET quoted\_identifier ON  
  
go  
  
CREATE TABLE [dbo].[pipeline\_parameter]  
  (  
     [parameter\_id]                       [INT] IDENTITY(1, 1) NOT NULL,  
     [server\_name]                        [NVARCHAR](500) NULL,  
     [src\_type]                           [NVARCHAR](500) NULL,  
     [src\_schema]                         [NVARCHAR](500) NULL,  
     [src\_db]                             [NVARCHAR](500) NULL,  
     [src\_name]                           [NVARCHAR](500) NULL,  
     [dst\_type]                           [NVARCHAR](500) NULL,  
     [dst\_name]                           [NVARCHAR](500) NULL,  
     [include\_pipeline\_flag]              [NVARCHAR](500) NULL,  
     [partition\_field]                    [NVARCHAR](500) NULL,  
     [process\_type]                       [NVARCHAR](500) NULL,  
     [priority\_lane]                      [NVARCHAR](500) NULL,  
     [pipeline\_date]                      [NVARCHAR](500) NULL,  
     [pipeline\_status]                    [NVARCHAR](500) NULL,  
     [load\_synapse]                       [NVARCHAR](500) NULL,  
     [load\_frequency]                     [NVARCHAR](500) NULL,  
     [dst\_folder]                         [NVARCHAR](500) NULL,  
     [file\_type]                          [NVARCHAR](500) NULL,  
     [lake\_dst\_folder]                    [NVARCHAR](500) NULL,  
     [spark\_flag]                         [NVARCHAR](500) NULL,  
     [dst\_schema]                         [NVARCHAR](500) NULL,  
     [distribution\_type]                  [NVARCHAR](500) NULL,  
     [load\_sqldw\_etl\_pipeline\_date]       [DATETIME] NULL,  
     [load\_sqldw\_etl\_pipeline\_status]     [NVARCHAR](500) NULL,  
     [load\_sqldw\_curated\_pipeline\_date]   [DATETIME] NULL,  
     [load\_sqldw\_curated\_pipeline\_status] [NVARCHAR](500) NULL,  
     [load\_delta\_pipeline\_date]           [DATETIME] NULL,  
     [load\_delta\_pipeline\_status]         [NVARCHAR](500) NULL,  
     [upsert\_key\_column]                  [NVARCHAR](500) NULL,  
     [incremental\_watermark\_column]       [NVARCHAR](500) NULL,  
     [incremental\_watermark\_value]        [DATETIME] NULL,  
     PRIMARY KEY CLUSTERED ( [parameter\_id] ASC )WITH (statistics\_norecompute =  
     OFF, ignore\_dup\_key = OFF) ON [PRIMARY]  
  )  
ON [PRIMARY]  
  
go

Now that you have created the pipeline\_parameter. Next, write a custom SQL query that will be used as the source of the ADF pipeline. Notice the addition of a SQLCommand and WhereValue which could be used to dynamically create a custom SQL statement and where clause based on whether the process type is full or incremental. For the purposes of this demo, only apply a filter for the incremental values, but the query demonstrates the flexibility to incorporate full loads into the same source query. Here’s our query:

SELECT src\_schema,  
       src\_db,  
       src\_name,  
       dst\_schema,  
       dst\_type,  
       dst\_name,  
       dst\_folder,  
       process\_type,  
       file\_type,  
       load\_synapse,  
       distribution\_type,  
       upsert\_key\_column,  
       incremental\_watermark\_column,  
       CASE  
         WHEN process\_type = 'FULL' THEN 'select \* from ' + src\_schema + '.' +  
                                         src\_name  
                                         + ' where  1 = '  
         WHEN process\_type = 'Incremental' THEN  
         'select \* from ' + src\_schema + '.' + src\_name  
         + ' where  ' + incremental\_watermark\_column  
         + ' > '  
       END                               AS SQLCommand,  
       CASE  
         WHEN process\_type = 'FULL' THEN '1'  
         WHEN process\_type = 'incremental' THEN Cast(  
         Isnull(incremental\_watermark\_value, 'yyyy-MM-dd') AS VARCHAR(50))  
       END                               AS WhereValue,  
       dst\_folder + '/' + dst\_name + '/' + file\_type + '/'  
       + Format(Getdate(), 'yyyy-MM-dd') AS FolderName,  
       dst\_name + '.' + file\_type        AS FileName  
FROM   dbo.pipeline\_parameter  
WHERE  load\_synapse = 1  
       AND process\_type = 'incremental'

Here is the SQL query that is contained in the SQLCommand column in Figure 13-1. This SQLCommand when combined with the WhereValue, dynamically forms the source SQL query that can be integrated and called as parameters within the ADF pipelines. The purpose of creating this process is to demonstrated the vast capabilities of building source SQL queries dynamically and then integrating it into an ADF pipeline

select \* from db.Log where CreatedDt >

Here is the SQL query that has been displayed in Figure 13-28.

SELECT Count(logid) AS IncrementalRecordCount

FROM [dbo].[log]

WHERE createddt > '04-01-2020'

Here is the SQL query that has been displayed in Figure 13-30.

SELECT Count(logid) AS IncrementalRecordCount

FROM [etl].[log]

WHERE createddt > '04-01-2020'

Here is the SQL query that has been displayed in Figure 13-31.

SELECT Count(logid) AS IncrementalRecordCount

FROM [dbo].[log]

WHERE createddt > '04-01-2020'

Here is the SQL query that has been displayed in Figure 13-34.

SELECT Count(logid) AS IncrementalRecordCount

FROM [etl].[log]

WHERE createddt > '04-01-2020'

Here is the SQL query that has been displayed in Figure 13-36.

SELECT Count(logid) AS IncrementalRecordCount

FROM [etl].[log]

WHERE facilityid = 100

Here are the SQL queries that are displayed in Figure 13-37:

SELECT Count(logid) AS IncrementalRecordCount

FROM [etl].[log]

SELECT Count(logid) AS IncrementalRecordCount

FROM [etl].[log]

WHERE facilityid = 100