This following code has been added to source query section of the ADF Lookup activity shown in Figure 7-1.

SELECT [id],  
       [server\_name],  
       [src\_type],  
       [src\_schema],  
       [src\_db],  
       [src\_name],  
       [dst\_type],  
       [dst\_name],  
       [include\_pipeline\_flag],  
       [partition\_field],  
       [process\_type],  
       [priority\_lane],  
       [pipeline\_date],  
       [pipeline\_status],  
       [load\_synapse],  
       [load\_frequency],  
       [dst\_folder],  
       [file\_type],  
       [lake\_dst\_folder],  
       [spark\_flag],  
       [data\_sources\_id],  
       [dst\_schema],  
       [distribution\_type],  
       [load\_sqldw\_etl\_pipeline\_date],  
       [load\_sqldw\_etl\_pipeline\_status],  
       [load\_sqldw\_curated\_pipeline\_date],  
       [load\_sqldw\_curated\_pipeline\_status],  
       [load\_delta\_pipeline\_date],  
       [load\_delta\_pipeline\_status]  
FROM   [dbo].[pipeline\_parameter]  
WHERE  load\_synapse = 1  
       AND pipeline\_status = 'success'  
       AND include\_pipeline\_flag = 1  
       AND process\_type = 'full'  
       AND load\_frequency = 'daily'

Here is the code that is used for the Items setting within the ForEach loop activity in Figure 7-3.

@activity('L\_Get\_Tables').output.value

The Names and Values used in the Source dataset properties in Figure 7-4 can be found below:

|  |  |
| --- | --- |
| Name | Value |
| dst\_name | @{item().dst\_name} |
| src\_schema | @{item().src\_schema} |
| distribution\_type | @{item().distribution\_type} |
| load\_sqldw\_etl\_pipeline\_date | @{item().load\_sqldw\_etl\_pipeline\_date} |
| load\_sqldw\_etl\_pipeline\_status | @{item().load\_sqldw\_etl\_pipeline\_status} |
| load\_sqldw\_curated\_pipeline\_date | @{item().load\_sqldw\_curated\_pipeline\_date} |
| load\_sqldw\_curated\_pipeline\_status | @{item().load\_sqldw\_curated\_pipeline\_status} |
| dst\_schema | @{item().dst\_schema} |

Here is the code that is used in the table connection setting in Figure 7-5.

etl.@{item().dst\_name}

Here is the code that is used in the table connection setting in Figure 7-6.

@{item().dst\_schema}.@{item().dst\_name}

Here is the code that is used in the sink Pre-copy script in Figure 7-7.

CREATE TABLE @{item().dst\_schema}.@{item().dst\_name}

WITH

(

CLUSTERED COLUMNSTORE INDEX,

DISTRIBUTION = @{item().distribution\_type}

)

AS SELECT TOP (0) \* FROM etl.@{item().dst\_name}

OPTION (LABEL = 'CTAS : @{item().dst\_name}');

Here is the code that is used in the sink Pre-copy script in Figure 7-8.

TRUNCATE TABLE @{item().dst\_schema}.@{item().dst\_name}

Here is the source code that has been used in the [etl].[Ctas\_from\_pipeline\_parameter] ADF stored procedure that is being called in the ForEach loop activity as shown in Figure 7-10.

SET ansi\_nulls ON  
  
go  
  
SET quoted\_identifier ON  
  
go  
  
CREATE PROC [etl].[Ctas\_from\_pipeline\_parameter] @schema            [VARCHAR](  
255),  
                                                 @name              [VARCHAR](  
255),  
                                                 @distribution\_type [VARCHAR](  
255)  
AS  
  BEGIN  
      DECLARE @table VARCHAR(255)  
      DECLARE @table\_stage VARCHAR(255)  
      DECLARE @table\_etl VARCHAR(255)  
      DECLARE @sql VARCHAR(max)  
  
      SET @table = @schema + '.' + @name  
      SET @table\_stage = @table + '\_stage'  
      SET @table\_etl = 'etl.' + @name  
      SET @sql = 'if object\_id (''' + @table\_stage  
                 + ''',''U'') is not null drop table '  
                 + @table\_stage + '; CREATE TABLE ' + @table\_stage  
                 + ' WITH ( DISTRIBUTION = ' + @distribution\_type  
                 + ' ,CLUSTERED COLUMNSTORE INDEX ) AS SELECT  \* FROM    ' + @table\_etl + '; if object\_id ('''  
                 + @table  
                 + ''',''U'') is not null drop table '  
                 + @table + '; RENAME OBJECT ' + @table\_stage + ' TO '  
                 + @name + ';'  
  
      EXEC(@sql)  
  END  
  
go

In a scenario where you might be interested in renaming the original curated table, rather than dropping the original curated table, use this script in the stored procedure activity within the ADF pipeline:

SET ansi\_nulls ON  
  
go  
  
SET quoted\_identifier ON  
  
go  
  
CREATE PROC [etl].[Ctas\_from\_pipeline\_parameter] @schema            [VARCHAR](  
255),  
                                                 @name              [VARCHAR](  
255),  
                                                 @distribution\_type [VARCHAR](  
255)  
AS  
  BEGIN  
      DECLARE @table VARCHAR(255)  
      DECLARE @table\_stage VARCHAR(255)  
      DECLARE @table\_drop VARCHAR(255)  
      DECLARE @table\_etl VARCHAR(255)  
      DECLARE @schematable\_drop VARCHAR(255)  
      DECLARE @sql VARCHAR(max)  
  
      SET @table = @schema + '.' + @name  
      SET @table\_stage = @table + '\_stage'  
      SET @table\_drop = @name + '\_drop'  
      SET @table\_etl = 'etl.' + @name  
      SET @schematable\_drop = @table + '\_drop'  
      SET @sql = 'if object\_id (''' + @table\_stage  
                 + ''',''U'') is not null drop table '  
                 + @table\_stage + '; CREATE TABLE ' + @table\_stage  
                 + ' WITH ( DISTRIBUTION = ' + @distribution\_type  
                 + ' ,CLUSTERED COLUMNSTORE INDEX ) AS SELECT  \* FROM    ' + @table\_etl + '; if object\_id ('''  
                 + @table  
                 + ''',''U'') is not null rename object '  
                 + @table + ' TO ' + @table\_drop + '; RENAME OBJECT '  
                 + @table\_stage + ' TO ' + @name + '; if object\_id ('''  
                 + @schematable\_drop  
                 + ''',''U'') is not null drop table '  
                 + @schematable\_drop + ';'  
  
      EXEC(@sql)  
  END  
  
go