The following script will create an updated version of the pipeline\_parameter table containing a few additional columns, along with the column parameter\_id as the primary key. Recall from our demonstrations in prior chapters that this table drives the meta-data ETL approach.

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,  
     PRIMARY KEY CLUSTERED ( [parameter\_id] ASC )WITH (statistics\_norecompute =  
     OFF, ignore\_dup\_key = OFF) ON [PRIMARY]  
  )  
ON [PRIMARY]  
  
go

This next script will create the pipeline\_log table for capturing the Data Factory success logs. In this table, column log\_id is the primary key and column parameter\_id is a foreign key with a reference to column parameter\_id from the table.

SET ansi\_nulls ON  
  
go  
  
SET quoted\_identifier ON  
  
go  
  
CREATE TABLE [dbo].[pipeline\_log]  
  (  
     [log\_id]                                [INT] IDENTITY(1, 1) NOT NULL,  
     [parameter\_id]                          [INT] NULL,  
     [datafactory\_name]                      [NVARCHAR](500) NULL,  
     [pipeline\_name]                         [NVARCHAR](500) NULL,  
     [runid]                                 [NVARCHAR](500) NULL,  
     [source]                                [NVARCHAR](500) NULL,  
     [destination]                           [NVARCHAR](500) NULL,  
     [triggertype]                           [NVARCHAR](500) NULL,  
     [triggerid]                             [NVARCHAR](500) NULL,  
     [triggername]                           [NVARCHAR](500) NULL,  
     [triggertime]                           [NVARCHAR](500) NULL,  
     [rowscopied]                            [NVARCHAR](500) NULL,  
     [dataread]                              [INT] NULL,  
     [no\_parallelcopies]                     [INT] NULL,  
     [copyduration\_in\_secs]                  [NVARCHAR](500) NULL,  
     [effectiveintegrationruntime]           [NVARCHAR](500) NULL,  
     [source\_type]                           [NVARCHAR](500) NULL,  
     [sink\_type]                             [NVARCHAR](500) NULL,  
     [execution\_status]                      [NVARCHAR](500) NULL,  
     [copyactivity\_start\_time]               [NVARCHAR](500) NULL,  
     [copyactivity\_end\_time]                 [NVARCHAR](500) NULL,  
     [copyactivity\_queuingduration\_in\_secs]  [NVARCHAR](500) NULL,  
     [copyactivity\_transferduration\_in\_secs] [NVARCHAR](500) NULL,  
     CONSTRAINT [PK\_pipeline\_log] PRIMARY KEY CLUSTERED ( [log\_id] ASC )WITH (  
     statistics\_norecompute = OFF, ignore\_dup\_key = OFF) ON [PRIMARY]  
  )  
ON [PRIMARY]  
  
go  
  
ALTER TABLE [dbo].[pipeline\_log]  
  WITH CHECK ADD FOREIGN KEY([parameter\_id]) REFERENCES  
  [dbo].[pipeline\_parameter] ([parameter\_id]) ON UPDATE CASCADE  
  
go

This next script will create a pipeline\_errors table which will be used to capture the Data Factory error details from failed pipeline activities. In this table, column error\_id is the primary key and column parameter\_id is a foreign key with a reference to column parameter\_id from the pipeline\_parameter table.

SET ansi\_nulls ON  
  
go  
  
SET quoted\_identifier ON  
  
go  
  
CREATE TABLE [dbo].[pipeline\_errors]  
  (  
     [error\_id]                    [INT] IDENTITY(1, 1) NOT NULL,  
     [parameter\_id]                [INT] NULL,  
     [datafactory\_name]            [NVARCHAR](500) NULL,  
     [pipeline\_name]               [NVARCHAR](500) NULL,  
     [runid]                       [NVARCHAR](500) NULL,  
     [source]                      [NVARCHAR](500) NULL,  
     [destination]                 [NVARCHAR](500) NULL,  
     [triggertype]                 [NVARCHAR](500) NULL,  
     [triggerid]                   [NVARCHAR](500) NULL,  
     [triggername]                 [NVARCHAR](500) NULL,  
     [triggertime]                 [NVARCHAR](500) NULL,  
     [no\_parallelcopies]           [INT] NULL,  
     [copyduration\_in\_secs]        [NVARCHAR](500) NULL,  
     [effectiveintegrationruntime] [NVARCHAR](500) NULL,  
     [source\_type]                 [NVARCHAR](500) NULL,  
     [sink\_type]                   [NVARCHAR](500) NULL,  
     [execution\_status]            [NVARCHAR](500) NULL,  
     [errordescription]            [NVARCHAR](max) NULL,  
     [errorcode]                   [NVARCHAR](500) NULL,  
     [errorloggedtime]             [NVARCHAR](500) NULL,  
     [failuretype]                 [NVARCHAR](500) NULL,  
     CONSTRAINT [PK\_pipeline\_error] PRIMARY KEY CLUSTERED ( [error\_id] ASC )WITH  
     (statistics\_norecompute = OFF, ignore\_dup\_key = OFF) ON [PRIMARY]  
  )  
ON [PRIMARY]  
textimage\_on [PRIMARY]  
  
go  
  
ALTER TABLE [dbo].[pipeline\_errors]  
  WITH CHECK ADD FOREIGN KEY([parameter\_id]) REFERENCES  
  [dbo].[pipeline\_parameter] ([parameter\_id]) ON UPDATE CASCADE  
  
go

the following script will create a stored procedure to update the pipeline\_log table with data from the successful pipeline run. Note that this stored procedure will be called from the Data Factory pipeline at run-time.

SET ansi\_nulls ON

go

SET quoted\_identifier ON

go

CREATE PROCEDURE [dbo].[usp\_updatelogtable] @datafactory\_name

VARCHAR(250),

@pipeline\_name

VARCHAR(250),

@runid

VARCHAR(250),

@source

VARCHAR(300),

@destination

VARCHAR(300),

@triggertype

VARCHAR(300),

@triggerid

VARCHAR(300),

@triggername

VARCHAR(300),

@triggertime

VARCHAR(500),

@rowscopied

VARCHAR(300),

@dataread

INT,

@no\_parallelcopies

INT,

@copyduration\_in\_secs

VARCHAR(300),

@effectiveintegrationruntime

VARCHAR(300),

@source\_type

VARCHAR(300),

@sink\_type

VARCHAR(300),

@execution\_status

VARCHAR(300),

@copyactivity\_start\_time

VARCHAR(500),

@copyactivity\_end\_time

VARCHAR(500),

@copyactivity\_queuingduration\_in\_secs

VARCHAR(500),

@copyactivity\_transferduration\_in\_secs VARCHAR(500)

AS

INSERT INTO [pipeline\_log]

([datafactory\_name],

[pipeline\_name],

[runid],

[source],

[destination],

[triggertype],

[triggerid],

[triggername],

[triggertime],

[rowscopied],

[dataread],

[no\_parallelcopies],

[copyduration\_in\_secs],

[effectiveintegrationruntime],

[source\_type],

[sink\_type],

[execution\_status],

[copyactivity\_start\_time],

[copyactivity\_end\_time],

[copyactivity\_queuingduration\_in\_secs],

[copyactivity\_transferduration\_in\_secs])

VALUES ( @datafactory\_name,

@pipeline\_name,

@runid,

@source,

@destination,

@triggertype,

@triggerid,

@triggername,

@triggertime,

@rowscopied,

@dataread,

@no\_parallelcopies,

@copyduration\_in\_secs,

@effectiveintegrationruntime,

@source\_type,

@sink\_type,

@execution\_status,

@copyactivity\_start\_time,

@copyactivity\_end\_time,

@copyactivity\_queuingduration\_in\_secs,

@copyactivity\_transferduration\_in\_secs )

go

Run the following script which will create a stored procedure to update the pipeline\_errors table with detailed error data from the failed pipeline run. Note that this stored procedure will be called from the Data Factory pipeline at run-time.

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[usp\_updateerrortable]

@datafactory\_name [nvarchar](500) NULL,

@pipeline\_name [nvarchar](500) NULL,

@runid [nvarchar](500) NULL,

@source [nvarchar](500) NULL,

@destination [nvarchar](500) NULL,

@triggertype [nvarchar](500) NULL,

@triggerid [nvarchar](500) NULL,

@triggername [nvarchar](500) NULL,

@triggertime [nvarchar](500) NULL,

@no\_parallelcopies [int] NULL,

@copyduration\_in\_secs [nvarchar](500) NULL,

@effectiveintegrationruntime [nvarchar](500) NULL,

@source\_type [nvarchar](500) NULL,

@sink\_type [nvarchar](500) NULL,

@execution\_status [nvarchar](500) NULL,

@errordescription [nvarchar](max) NULL,

@errorcode [nvarchar](500) NULL,

@errorloggedtime [nvarchar](500) NULL,

@failuretype [nvarchar](500) NULL

AS

INSERT INTO [pipeline\_errors]

(

[datafactory\_name],

[pipeline\_name],

[runid],

[source],

[destination],

[triggertype],

[triggerid],

[triggername],

[triggertime],

[no\_parallelcopies],

[copyduration\_in\_secs],

[effectiveintegrationruntime],

[source\_type],

[sink\_type],

[execution\_status],

[errordescription],

[errorcode],

[errorloggedtime],

[failuretype]

)

VALUES

(

@datafactory\_name,

@pipeline\_name,

@runid,

@source,

@destination,

@triggertype,

@triggerid,

@triggername,

@triggertime,

@no\_parallelcopies,

@copyduration\_in\_secs,

@effectiveintegrationruntime,

@source\_type,

@sink\_type,

@execution\_status,

@errordescription,

@errorcode,

@errorloggedtime,

@failuretype

)

GO

The following values listed in Table 9-1 will need to be entered into the Stored Procedure activity parameter values shown in Figure 9-9.

Table 9-1. List of values to be entered into the Stored Procedure activity’s parameters settings

|  |  |
| --- | --- |
| Name | Values |
| DataFactory\_Name | @{pipeline().datafactory} |
| Pipeline\_Name | @{pipeline().pipeline} |
| RunId | @{pipeline().runid} |
| Source | @{item().src\_name} |
| Destination | @{item().dst\_name} |
| TriggerType | @{pipeline().triggertype} |
| TriggerId | @{pipeline().triggerid} |
| TriggerName | @{pipeline().triggername} |
| TriggerTime | @{pipeline().triggertime} |
| rowsCopied | @{activity('Copy-Table').output.rowscopied} |
| RowsRead | @{activity('Copy-Table').output.rowsread} |
| No\_ParallelCopies | @{activity('Copy-Table').output.usedparallelcopies} |
| copyDuration\_in\_secs | @{activity('Copy-Table').output.copyduration} |
| effectiveIntegrationRuntime | @{activity('Copy-Table').output.effectiveintegrationruntime} |
| Source\_Type | @{activity('Copy-Table').output.executiondetails[0].source.type} |
| Sink\_Type | @{activity('Copy-Table').output.executiondetails[0].sink.type} |
| Execution\_Status | @{activity('Copy-Table').output.executiondetails[0].status} |
| CopyActivity\_Start\_Time | @{activity('Copy-Table').output.executiondetails[0].start} |
| CopyActivity\_End\_Time | @{utcnow()} |
| CopyActivity\_queuingDuration\_in\_secs | @{activity('Copy-Table').output.executionDetails[0].detaileddurations.queuingDuration} |
| CopyActivity\_transferDuration\_in\_secs | @{activity('Copy-Table').output.executionDetails[0].detaileddurations.transferDuration} |

The following values will need to be entered into the stored procedure parameter values as shown in Figure 9-11.

Table 9-2. List of values to be entered into the Stored Procedure activity’s parameters settings

|  |  |
| --- | --- |
| Description | Source |
| DataFactory\_Name | @{pipeline().datafactory} |
| Pipeline\_Name | @{pipeline().pipeline} |
| RunId | @{pipeline().runid} |
| Source | @{item().src\_name} |
| Destination | @{item().dst\_name} |
| TriggerType | @{pipeline().triggertype} |
| TriggerId | @{pipeline().triggerid} |
| TriggerName | @{pipeline().triggername} |
| TriggerTime | @{pipeline().triggertime} |
| No\_ParallelCopies | @{activity('Copy-Table').output.usedparallelcopies} |
| copyDuration\_in\_secs | @{activity('Copy-Table').output.copyduration} |
| effectiveIntegrationRuntime | @{activity('Copy-Table').output.effectiveintegrationruntime} |
| Source\_Type | @{activity('Copy-Table').output.executiondetails[0].source.type} |
| Sink\_Type | @{activity('Copy-Table').output.executiondetails[0].sink.type} |
| Execution\_Status | @{activity('Copy-Table').output.executiondetails[0].status} |
| ErrorCode | @{activity('Copy-Table').error.errorcode} |
| ErrorDescription | @{activity('Copy-Table').error.message} |
| ErrorLoggedTIme | @utcnow() |
| FailureType | @concat(activity('Copy-Table').error.message,'failuretype:',activity('Copy-Table').error.failuretype) |