Loading Data into Azure DW with ADF

Technologies showcased: ADF GUI, Azure DW, Polybase, Azure DB

Author: Joshuha Owen

Date: 1/3/2018

Table of Contents

[Summary 3](#_Toc502921583)

[Prerequisites 4](#_Toc502921584)

[Scenario 4](#_Toc502921585)

[Part 1 – Creating the Pipeline for the Data Warehouse Load 4](#_Toc502921586)

[Part 2 – Creating the ODS to DW Staging copies 9](#_Toc502921587)

[Part 3 – Creating the Azure Blob to DW Staging copies 19](#_Toc502921588)

[Part 4 – Calling the Stored Procedure to Load the Staging Data into the Azure Data Warehouse 28](#_Toc502921589)

## Summary

In this Lab Module, we will walk through loading an Azure Data Warehouse database with information from our Azure ODS database as well as flat files generated for our weather data and the output of a Hive query (completed in previous lab modules). After we copy the data to staging tables in the Azure DW database we will be using ADF to call stored procedures to load the dimensions and the fact table. This involves:

* Creating a Stored Procedure activity to truncate our staging tables
* Creating Copy activities to copy Azure DB and Azure Blob files to the staging schema
* Creating Stored Procedure activities to call a load dimensions and load fact stored procedure on the Azure DW database

## Prerequisites

* Azure Subscription with rights to use/deploy Azure services, and X of Azure credit
* Azure Blob Storage container
* Azure SQL Database
* Azure SQL Data Warehouse
* Azure Data Factory

## Scenario

|  |  |  |
| --- | --- | --- |
| Part 1 – Creating the Pipeline for the Data Warehouse Load | | |
| **Scenario** | | |
| We will set up a new pipeline to run our data warehouse load pattern into our staging area in the Azure DW database. We will source the data from both our Azure ODS Database as well as Azure Blob files created in previous lab modules (Lab modules 3 through 5). | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
| We will use the Azure Data Factory we created in Lab 01 called adflab-adf.. | 1. Navigate to the Azure portal within your web browser and navigate to <https://portal.azure.com> 2. Open the Azure Data Factory blade [adflab-adf], pinned from a previous lab (or navigate to it using the All Resources menu item). |  |
|  | 1. In the Overview blade you should see the following Quick Links: |  |
|  | 1. Click the Pipeline Editor button and you should see graphical user interface Overview page. |  |
|  | 1. Click the Create Pipeline button under Let’s Get Started. |  |
|  | 1. You should now see Pipeline GUI editor. |  |
|  | 1. Enter Merge FAA Files as the Pipeline Name. |  |
|  | 1. In the Pipeline GUI, drag the Stored Procedure activity to the empty pane above General. 2. Rename the activity to DW Truncate Staging. |  |
|  | 1. Click the SQL Account tab. 2. Click the +New button next to the Linked Service dropdown. |  |
| As part of the setup script, the DWLoadUser was created and assigned the LargeRC role. This follows best practices on not using admin account for data load patterns. The password is the same as the admin account defined in Lab Module 1.  <https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-develop-concurrency> | 1. You should now see the New Linked Service configuration pane. 2. Enter the following information: Name: Azure DW DB Type: Azure SQL Data Warehouse Account Selection Method: From Azure Subscription Azure Subscription: Your Azure Subscription Server Name: Your SQL Server created in Lab module 1 Database Name: AirlinePerformance-DW User name: DWLoadUser Password: Password you defined in the PowerShell Deployment script in Lab module 1. The script default is L@bP@ss01 3. Click the Test connection button. 4. Click the Save button. |  |
| This stored procedure was part of the setup scripts in Lab Module 01 for the AirlinePerformance-DW. This procedure truncates each table in the Staging schema. | 1. Click the Stored Procedure tab. 2. For the Stored procedure name input usp\_TruncateStaging 3. Click the Publish button to save your Pipeline. |  |

|  |  |  |
| --- | --- | --- |
| Part 2 – Creating the ODS to DW Staging copies | | |
| **Scenario** | | |
| We will create Copy activities to copy files from our Azure ODS DB to the Azure SQL DW DB Staging tables. This process is the same for each ODS table (3 tables) and the steps will be repeated for each copy activity. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
| Note in Step 4 the names will be the following for each table copy:  ODS Airline to DW  ODS Airport to DW  ODS Flight to DW | 1. In the Pipeline GUI drag the Copy activity to the right of the Stored Procedure activity. 2. Drag from the Green box on the right of the Stored Procedure activity to the Copy activity to create a Success constraint. 3. Click the Copy activity. 4. Name the Activity as ODS Airline to DW. |  |
|  | 1. Click the Source tab. 2. Click the +New button next to the Source Dataset dropdown. |  |
|  | 1. Select Azure SQL Database and click Finish. |  |
|  | 1. You should see the Dataset configuration page. 2. For the Name input ODS Airline. |  |
|  | 1. Click the Connection tab. 2. If this is the first time creating the Linked Service, click the +New button next to Linked service. If you are repeating this step for other tables, select the Azure ODS DB linked service. | OR |
| **Only perform this step for the first table created if you clicked the +New button.** | 1. You should see the new Linked Service configuration pane. 2. Enter the following information: Name: Azure ODS DB Type: Azure SQL Database Account Selection Method: From Azure Subscription Azure Subscription: Your Azure Subscription Server Name: Your SQL Server created in Lab module 1 Database Name: AirlinePerformance-ODS User name: SQL user you defined in the PowerShell Deployment scrip in Lab module 1. The script default is labadmin. Password: Password you defined in the PowerShell Deployment script in Lab module 1. The script default is L@bP@ss01 3. Click the Test connection button. 4. Click Save. |  |
| When repeating this step use the following tables:  dbo.Airline  dbo.Airport  dbo.Flight | 1. For the Table dropdown select [dbo].[Airline] |  |
| After this step the dataset should ready and you can return to your Copy Activity to continue. | 1. Click the Schema tab. 2. Click the Import Schema button. 3. Click Save. |  |
| Note when repeating this activity, you will see the corresponding dataset you created in the previous steps. | 1. Click your Load Data Warehouse pipeline in the left menu to return to the Pipeline GUI screen. 2. Click the ODS Airline to DW Copy activity. 3. Click the Source tab. Your Source Dataset should now be set to ODS Airline. |  |
|  | 1. Click the Sink Tab. 2. Click the +New button next to the Sink Dataset dropdown. |  |
|  | 1. Select Azure SQL Data Warehouse and click the Finish button. |  |
| When you repeat these steps use the following dataset names:  DW Staging Airline  DW Staging Airport  DW Staging Flight | 1. You should now the see Azure SQL Data Warehouse dataset configuration screen. 2. For the Name input DW Staging Airline. 3. Click the Connection tab. |  |
|  | 1. In the Linked Service dropdown select the Azure DW DB linked serviced created in earlier sections of this lab module. 2. In the Table dropdown select [Staging].[DimAirline]. 3. Click the Schema tab. |  |
| Note after hitting Save you have finished creating your Sink dataset. | 1. Click the Import Schema button. 2. Click the Save icon at the top. |  |
| Note when repeating this activity, you will see the corresponding dataset you created in the previous steps. | 1. Click your Load Data Warehouse pipeline in the left menu to return to the Pipeline GUI screen. 2. Click the ODS Airline to DW Copy activity. 3. Click the Sink tab. Your Sink Dataset should now be set to DW Staging Airline. 4. Click the Polybase Settings expansion. 5. Check the box Allow Polybase. 6. Click the Mapping tab. |  |
|  | 1. Click the Import Schema button. 2. Verify the field names are matched to the correct Field. If not, manually correct them in the dropdown menus. 3. Click the Settings tab. |  |
| For Azure SQL DW to copy the data, it needs to stage the data to a temporary location and either use Polybase (which we set earlier) or a Bulk Insert. In this step, we are configuring the temporary location to host those files. | 1. Click the Enable Staging True radio selection. 2. In the Staging Account Linked Service select the AZStorage-Staging linked service created in Lab Module 3. 3. Under Storage Path click the Browse button. Select the output folder and click Finish. 4. Click the Publish button to save your Pipeline. |  |
|  | 1. Repeat the above steps so that you have 3 ODS to DW Copy activities. The final result should look as follows. |  |

|  |  |  |
| --- | --- | --- |
| Part 3 – Creating the Azure Blob to DW Staging copies | | |
| **Scenario** | | |
| We will be creating Copy activities to copy files from our Azure Blob storage to the Azure SQL DW Staging tables. This process is the same for each file (2 files) and the steps will be repeated for each copy activity. These files were created as outputs for Lab Modules 4 and 5. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
| Note in Step 4 the names will be the following for each file copy:  Blob FAA to DW  Blob Weather to DW | 1. In the Pipeline GUI, drag the Copy activity to the right of the Stored Procedure activity. 2. Create a Success Constraint by dragging from the Green box on the right of the Stored Procedure activity to the Copy activity. 3. Click the Copy activity. 4. Name the Activity to Blob FAA to DW. |  |
| Note for the Weather data file, this dataset was created as part of Lab 4 and you can select Weather Data Blob and skip to Step 17. | 1. Click the Source tab. 2. Click the +New button next to the Source Dataset. |  |
|  | 1. Click Azure Blob Storage and then the Finish button. |  |
| This file is an output from the HDInsight Hive query that joined together Aircraft information of the plane between two FAA files. | 1. You should see the Azure Blob Storage dataset configuration page. 2. For the name input FAA Merge. 3. Click the Connection tab. |  |
|  | 1. Fill out the following information: Linked Service: Select AzStorage-Staging File Path: Browse to the output/FAAMerge folder and click the 000000\_0 file. 2. Verify Source files contain column names in first row is checked. 3. Click the Schema tab. |  |
|  | 1. Click Import Schema. 2. Click the Save Icon at the top. |  |
| Note when repeating this activity, you will see the corresponding dataset you created in the previous steps. | 1. Click your Load Data Warehouse pipeline in the left menu to return to the Pipeline GUI screen. 2. Click the Blob FAA to DW activity. 3. Click the Source tab. Your Source dataset should now be set to FAA Merge. |  |
|  | 1. Click the Sink Tab. 2. Click the +New button next to the Sink Dataset dropdown. |  |
|  | 1. Select Azure SQL Data Warehouse and click the Finish button. |  |
| When you repeat these steps use the following dataset names:  DW Staging Aircraft  DW Staging Weather | 1. You should now the see Azure SQL Data Warehouse dataset configuration screen. 2. For the Name input DW Staging Aircraft 3. Click the Connection tab. |  |
|  | 1. In the Linked Service dropdown, select the Azure DW DB Linked Service created in earlier sections of this lab module. 2. In the Table dropdown select [Staging].[DimAircraft]. 3. Click the Schema tab. |  |
| Note after hitting Save you will be finished creating your Sink dataset. | 1. Click the Import Schema button. 2. Click the Save icon at the top. |  |
| Note when repeating this activity, you will see the corresponding dataset you created in the previous steps. | 1. Click your Load Data Warehouse pipeline in the left menu to return to the Pipeline GUI screen. 2. Click the Blob FAA to DW activity. 3. Click the Sink tab. Your Sink dataset should now be set to DW Staging Aircraft. 4. Click the Polybase Settings expansion. 5. Check the box Allow Polybase. 6. Click the Mapping tab. |  |
|  | 1. Click the Import Schema button. 2. Verify the field names are matched to the correct Field. If not, manually correct them in the dropdown menus. 3. Click the Settings tab. |  |
| For Azure SQL DW to copy the data, it needs to stage the data to a temporary location and either use Polybase (which we set earlier) or a Bulk Insert. In this step, we are configuring the temporary location to host those files. | 1. Click the Enable Staging True radio selection. 2. In the Staging Account Linked Service select the AZStorage-Staging linked service created in Lab Module 3. 3. Under Storage Path click the Browse button. Select the output folder and click Finish. 4. Click the Publish button to save your Pipeline.. |  |
|  | 1. Repeat the above steps so that you have 2 Blob to DW Copy activities. The final result should look as follows. |  |

|  |  |  |
| --- | --- | --- |
| Part 4 – Calling the Stored Procedure to Load the Staging Data into the Azure Data Warehouse | | |
| **Scenario** | | |
| We will connect all the copy activities to a stored procedure to load the dimensions and then load our fact table. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
|  | 1. In the Pipeline GUI drag the Stored Procedure activity to the right of all the Copy Activities. 2. Create a Success Constraint by dragging from the Green box on the right of the Stored Procedure activity to the Copy activity. 3. Click the Stored Procedure Activity. 4. Name the Activity as DW Load Dimensions |  |
|  | 1. Click the SQL Account tab. 2. In the Linked Service dropdown select the Azure DW DB linked service. |  |
| This stored procedure was part of the setup scripts in Lab Module 01 for the AirlinePerformance-DW. This procedure runs an upsert pattern from Staging to the ADF schema by using a rename pattern since MERGE is not supported in Azure SQL Data Warehouse.  [https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-develop-ctas#replace-merge-statements](https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-develop-ctas%23replace-merge-statements) | 1. Click the Stored Procedure tab. 2. For the Stored Procedure name input usp\_LoadDimensions. |  |
|  | 1. In the Pipeline GUI drag the Stored Procedure activity to the right the DW Load Dimensions activity. 2. Drag from the Green box on the right of the DW Load Dimensions to the new Stored Procedure Activity. 3. Click the Stored Procedure Activity. 4. Name the Activity to DW Load Fact |  |
|  | 1. Click the SQL Account tab. 2. In the Linked Service dropdown select the Azure DW DB linked service. |  |
| This stored procedure was part of the setup scripts in Lab Module 01 for the AirlinePerformance-DW. This procedure runs checks for the max flight date in the existing Fact table (ADF.FlightFact) and pulls any records from the Staging.Flight table that are newer than this date. It then attempts to lookup the dimension key values based on the business code in each dimension (and uses -1 if it can’t be located). | 1. Click the Stored Procedure tab. 2. For the Stored Procedure name input usp\_LoadFact. 3. Click the Publish button to save your Pipeline. |  |
|  | 1. The Load Data Warehouse pipeline is ready to be run. You can click on Test Run at this point to verify each activity runs correctly or look at Lab Module 7 and 8 where will be integrating this pipeline into a Master Pipeline and running a schedule trigger. |  |

**IMPORTANT: AVOID INCURRING EXTRA CHARGES BY PAUSING YOUR SUBSCRIPTION RESOURCES**