Lab 8 - Activities

Requirements

To be able to start the lab, it's important that Lab5 has been completed.

Objective

We have already used several activities such as Copy, Web, Wait, and Set variable. There are many more, and some are very handy when you make them work together in order to create advanced pipelines. Follow the assignments step by step.

Assignment 1 - Execute Stored Procedure

Stored Procedures are programs stored on the database. Often actions on the database (such as emptying a table, or starting a process within the database) are "captured" in a stored procedure. With ADF, you can now orchestrate these.

1. Ensure that you're back in the **not** linked ADF. Click on **Pipeline Actions** in Pipelines, and on **New Pipeline**.
2. Name the pipeline: `PL_Process_Dates_Training`.
3. From the list of **Activities**, click on the **General** option. Click and drag **Stored Procedure** onto the canvas.
4. Name the **Stored Procedure** as follows: **USP_DL_Dates**. Then click on the **Settings** tab.
5. Choose the `LS_sqldb_target` for **Linked service** and for **Stored Procedure name**, choose **[Stg].[USP_DL_Dates]**.
6. Click on **Stored procedure parameters** and then on **import**. The parameters of the stored procedure will now be loaded.
7. Enter the following in **StartYear**: **1900**.
8. Enter the following in **EndYear**: **2099**.
9. Click on the **Blue button** with the text **Publish all** and then on the **Publish** button.
10. Click on **Debug** and wait until the pipeline is ready.

Assignment 2 - Conditional Filtering

You can also use data from the database to perform your orchestration, for example:

- A process that can only start if a certain row is present in your settings table
- Starting a separate pipeline for each customer present in the Customers table

Firstly, we retrieve data from a SQL database here, and apply a filter to that data within the ADF pipeline.

1. Click on **Dataset Actions** in Datasets, and on **New Dataset**.
2. Search for **SQL** and choose **Azure SQL Database**. Then click on **Continue**.
3. Name the Dataset as follows: `DS_asql_SalesLT_Customers_Training` and choose `LS_sqldb_source` as the linked service.
4. Choose **SalesLT.Customer** for **Table** and click on **OK**.
5. Click on **Pipeline Actions** in Pipelines, and on **New Pipeline**.
6. Name the pipeline: `PL_Filter_SalesPersonal_Training`.
7. From the list of **Activities**, click on the **General** option. Click and drag **Lookup** onto the canvas.
8. Name the lookup: **Lookup_SalesPersonal**.
9. Go to the **Settings** tab and choose the `DS_asql_SalesLT_Customers_Training` for **Source dataset**.
10. Uncheck the box for **First row only**.
11. Click on the **Use query** option for **query** and type or paste the following code:

    ```
    SELECT
    COUNT(*) AS Registered_Customers,
    SalesPerson

    FROM [SalesLT].[Customer]

    GROUP BY SalesPerson
    ```

12. From the list of **Activities**, click on the **Iteration & conditionals** option. Click and drag **filter** onto the canvas.
13. Drag the **green block** from the Lookup to the filter activity. So they are connected to each other.
14. Name the **filter** as **Best seller**.
15. Click on the **Settings** tab and click next to **Items** and then on **Add dynamic content**.
16. Under **Activity outputs** click on **Lookup_SalesPersonal value array** and click on **OK**.
17. Click next to **Condition** and then on **Add dynamic content**.
18. Go to **Functions**, and type or paste the following code: `@greaterOrEquals(item().Registered_Customers,100)`
19. Click on the **Blue button** with the text **Publish all** and then on the **Publish** button.
20. Click on **Debug** and wait until the pipeline is ready, view the results by looking at the **Output** of the **Best seller** step.

Assignment 3 - Inserting into a Stored Procedure

Within ADF there is the possibility for a **stored procedure insert**. This allows you to add logic on the database side about how incoming data should be treated. If you're curious how this works, you can connect to the target database with, for example, Azure Data Studio or SSMS. You can then view the stored procedure definition `Stg.USP_DL_SalesOrderHeader`.

1. Click on **Dataset Actions** in Datasets, and on **New Dataset**.
2. Search for **SQL** and choose **Azure SQL Database**. Then click on **Continue**.
3. Name the Dataset `DS_asql_SalesLT_SalesOrderHeader_Training` and choose `LS_sqldb_source` as the linked service.
4. Choose **SalesLT.SalesOrderHeader** for **Table name** and click on **OK**.
5. Hover your mouse over `DS_asql_SalesLT_SalesOrderHeader_Training` and click on the **3 dots behind it** (Actions).
6. Then click on the **Clone** option, a copy of the Dataset will appear.
7. Rename this Dataset to `DS_asql_Stg_SalesOrderHeader_Training` and change the linked service to `LS_sqldb_target`.
8. Click on **Edit** under **Table**.
    - Empty the first field (schema)
    - Type or paste `DeltaTable` in the second field (table name).
9. Click on **Pipeline Actions** in Pipelines, and on **New Pipeline**.
10. Name the Pipeline as follows: `PL_copy_Deltaload_SalesOrderHeader_Training`.
11. From the list of **Activities**, click on the **Move & transform** option. Click and drag **Copy Data** onto the canvas.
12. Name the **Copy data** as `Copy SalesOrderHeader`. Then click on the **Source** tab and choose `DS_asql_SalesLT_SalesOrderHeader_Training` for the **Source dataset**.
13. Click on the **Sink** tab and choose `DS_asql_Stg_SalesOrderHeader_Training` for the **Sink dataset**.
14. Enter the following code in **Pre-copy Script**: `Truncate table [Stg].[SalesOrderHeader]`.
15. Choose **Stored Procedure** for **Write behavior** and select here `[Stg].[USP_DL_SalesOrderHeader]`.
16. Change the **Table type** to **[Stg].[UDT_SalesOrderHeader]**.
17. Ensure the **Table type parameter name** is set to **Deltatable**.
18. Click on the **Blue button** with the text **Publish all** and then on the **Publish** button.
19. Click on **Debug** and wait until the pipeline is ready.

Assignment 4 - Dynamic Pipelines/Datasets

So far, we have loaded all the tables one by one, with their own datasets and pipelines. However, this is not necessary in ADF: you can make your pipelines and datasets *dynamic*. This means:

- You add parameters to your dataset (for example, for the table name)
- You leave the *schema* empty. ADF now performs a *schema infer*, which means that the schema is determined at the moment of execution.
- When using the dataset, you pass the required parameters.

This way, you can, for example, read a list of tables to retrieve from a CSV file or SQL configuration table, after which you read them one by one with a ForEach loop.

1. Click on **Dataset Actions** in Datasets, and on **New Dataset**.
2. Search for **SQL** and choose **Azure SQL Database**. Then click on **Continue**.
3. Name the Dataset as follows: `DS_asql_sqldb_SourceTables_Training` and choose `LS_sqldb_source` as the linked service.
4. Leave the **Table name** empty and click on **OK**.
5. Repeat steps 1 to 4 for the **sqldb-target** and name the Dataset as follows: `DS_asql_sqldb_TargetTables_training`.
6. When the Dataset for the **sqldb-target** is created, go to the **Parameters** tab and click on **New**.
7. Name the parameter `TargetTableName`.
8. Go to the **Connection** tab and check **Edit**. Enter **Stg** in the first field and click on the 2nd field, then on **Add dynamic content**.
9. From the list, choose the parameter named: **TargetTableName** and then click on **OK**.
10. Click on **Pipeline Actions** in Pipelines, and on **New Pipeline**.
11. Name the Pipeline as follows: `PL_copy_deltaload_Training`.
12. From the list of **Activities**, click on the **General** option. Click and drag **Lookup** onto the canvas.
13. Name the **lookup** `Lookup_SourceTables`, then click on the **Settings** tab and choose `DS_aqsl_sqldb_SourceTables_training` as the **Source dataset**.
14. For **Use query**, select the **Query** option. Then click in the query field and paste or type the following query:

    ```
    SELECT
    TABLE_SCHEMA AS Table_Schema,
    TABLE_NAME AS Table_Name
    FROM INFORMATION_SCHEMA.TABLES
    WHERE TABLE_SCHEMA = 'SalesLT' AND TABLE_TYPE = 'BASE TABLE'
    ```

    If **First row only** is checked, uncheck it.

15. From the list of **Activities**, click on the **Iteration & Conditionals** option. Click and drag **ForEach** onto the canvas.
16. Drag the **Green block** from **Lookup_SourceTables** to the **ForEach** so they are sequentially connected.
17. Name the **ForEach** `ForEachTable` and click on the **Settings** tab.
18. Click on the area next to **Items**, then on **Add dynamic content**. Choose `Lookup_SourceTables value array`.
19. Click on the **plus** in the **ForEachTable**. Now choose the activity **Copy Data**.
20. Name the **Copy data** `Copy Tables`, then click on the **Source** tab and choose `DS_aqsl_sqldb_SourceTables_training` as the **Source dataset**.
21. For **Use query**, select the **Query** option. Then click in the query field, then on **Add dynamic content** and type or paste:

    ```
    SELECT * FROM @{item().Table_Schema}.@{item().Table_Name}
    ```

22. Click on the **Sink** tab and then choose the `DS_aqsl_sqldb_TargetTables_training` linked service, then click on the field next to **TargetTableName**, followed by **Add dynamic content** and paste or type:

    ```
    @item().Table_Name
    ```

23. Click on the field next to **Pre-copy script**, then on **Add dynamic content** and paste or type:

    ```
    Truncate Table Stg.@{item().Table_Name}
    ```

24. Click on the **Blue button** with the text **Publish all** and then on the **Publish** button.
25. Click on **Debug** and wait until the pipeline is ready.

End of Lab 8

Table of Contents