**Data wrangling project**

**Problem statement**:

Your company, Globomantics, is creating a data solution that gathers movie titles as well as their ratings and streams them into an Azure Blob Storage container for further analysis. Globomantics' Developer Team is skilled in SQL and wants to autoscale the application as needed while maintaining costs and development efforts to the minimum; therefore they have set up an Azure CosmosDB SQL database for further usage. You are in charge of developing the pipelines within Azure Data Factory to extract the movie data from Blob Storage, cleanse it by removing duplicates and Null values, and load it into Azure CosmosDB.

1. From the Azure Data Factory canvas, on the left panel, click on Move & Transform and drag and drop the Data flow box into the canvas.

Note: Data flow is a transformation tool that allows scripting and has several cookbook recipes to perform data cleansing

1. Click on the Data flow box you've just dragged into the canvas, click on the Settings tab and select + New. This will start a new Data flow canvas that Data Factory will run when the pipeline gets triggered.

Note: You may think this is a useless abstraction, but Pipelines are usually excellent to integrate with other Azure services while Data flow permits real-time debugging of transformation scripts.

You may be prompted with a tutorial dialogue. Click on Finish

1. Click on Add Source, while avoiding the extra dialogues that pop up. A Source represents incoming Data from a Data entity.
2. Under the Source settings panel on the bottom, click the + New button to create a Dataset.

Note: A Dataset is a representation of the Data flowing into or out of the Pipeline. A key aspect of Data Factory is that data still resides in its location for its execution.

1. In the New dataset panel, click on Azure Blob Storage > Continue > DelimitedText > Continue.

Note: Afterwards, you may check connectors exist for many AWS and GCP sources as well as Azure.

1. On the same panel, click the Linked service dropdown and select + New to create a new Linked Service that holds the dataset.
2. A New linked service panel pops up, leave all the options as default and on the Storage account name drop-down, select the only available storage account, which will have a randomly generated name . Next, click on Create.

Note: At this point, the New linked service panel will close itself and the Dataset panel will pop up again to finish configuring it.

1. Fill out the Set properties panel as follows:

Linked service: AzureBlobStorage1.

Type source in the Container text box, and type source.csv in the File text box.

Check the option First row as header

Click OK.

If you check the source box in the canvas, it says Columns: 1 total, to review that we will start a Debug session.

1. On the top of the canvas, click the slider Data flow debug and click OK in the right panel for the debug cluster configuration; it will take around 5 minutes to start the debug cluster. Once started, Data preview is available to inspect the state of the data at each step of the pipeline.
2. In the same bottom Dataset panel, click on Data preview > Refresh. We can see that the different columns are separated by semicolons (;) but the default for CSV is commas (,), therefore compacting all into one column.
3. Click the Source settings tab of the Dataset panel, click on Open and set the column delimiter as Semicolon (;)
4. 1. Finally, in the Schema tab, click on Import schema > From connection/store.If you go to the dataflow1 canvas again, as the image shows, the Source box source1 should have Columns: 3 total:

A screenshot of a computer

Description automatically generated

**Delete Duplicate Rows in Data flow**

﻿

1. From the canvas, click on the Script papyrus icon on the top right, as the image shows. This will show the code that the canvas is generating; one can work in Data flow with the script avoiding the UI as well.

﻿

﻿A screenshot of a computer

Description automatically generated

1. Copy and paste the following snippet on the bottom of the script:

source1 aggregate(groupBy(mycols = sha2(256,columns())),

each(match(true()), $$ = first($$))) ~> DistinctRows

Click OK.



**Split Pipeline Data Based on Null Conditions**

1. From the Data flow canvas, click on the Script papyrus icon.
2. Copy and paste into the end of the script the following snippet:

DistinctRows split(contains(array(columns()),isNull(#item)),

disjoint: false) ~> LookForNULLs@(hasNULLs, noNULLs)

Click the OK button.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Note: This snippet checks if any column has Null values and splits the rows incoming into the step based on that condition.

1. Over the left panel, on Datasets, hover over the number 1 and click on the ellipsis ( ... ) > New dataset

Click on Azure Blob Storage > Continue > DelimitedText > Continue.

Fill out the Set properties panel as follows:

Linked service: AzureBlobStorage1.

Type failed in the Container text box.

Check the option First row as header

Click OK.

1. Repeat step 3 to create a new Dataset. Select Azure Cosmos DB (SQL API) > Continue and from the Linked service drop down select + New.
2. Fill out the New linked service panel as follows:

Azure Cosmos DB account name: select the only option, which should begin with cosmosdb, followed by a random string.

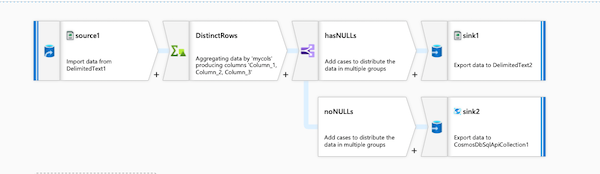
Database name: select the sink database.

Leave all other fields with default values.

Click the Create button.

In the Collection dropdown select movies and click OK.

1. Back in the dataflow1 canvas again, click the + icon on the bottom right section of the box hasNULLs. Select Sink under the Destination section. On the Dataset dropdown on the bottom panel select DelimitedText2.
2. On the bottom right section of the box, noNULLs click the + icon. Select Sink under the Destination section. On the Dataset dropdown on the bottom panel select CosmosDbSqlApiCollection1.
3. At this point, you should have a Pipeline like the following image below:



References:

<https://learn.microsoft.com/en-us/azure/data-factory/data-flow-script>

<https://learn.microsoft.com/en-us/azure/data-factory/author-visually?tabs=data-factory>

﻿