Hands-On Lab: Load Multiple CSV Files into Azure SQL Database with File Names Using Mapping Data Flow

Author: Sandeep Kumar Sharma

Learning Objective

This lab teaches you how to load **multiple CSV files** from Azure Storage into a **single Azure SQL Database table**, while also capturing the **file name** of each CSV file inside the Mapping Data Flow. The lab is designed in a very simple, step-by-step manner for beginners.

Learning Outcome

After completing this lab, you will be able to: - Read multiple CSV files using wildcard paths - Extract each file name inside Mapping Data Flow - Add the file name as a column to every record - Load the final combined data into Azure SQL Database

---

1. Introduction

Real-time data ingestion often involves multiple CSV files placed inside a folder. Each file may represent daily data, region-wise data, or separate business units. While loading these files into a SQL table, it is extremely useful to store the **file name** to identify where each record came from.

This lab will help you extract and store file names along with row data.

---

2. Prerequisites

Before starting: - You have an Azure Data Factory instance - You have a folder containing multiple CSV files (example: `/input/`) in ADLS/Blob - You have an Azure SQL Database with a table created

Example CSV files:

```
/input/sales_jan.csv
/input/sales_feb.csv
/input/sales_mar.csv
```

Each file contains:

```
Product,Quantity,Price
Pen,10,20
Book,5,100
```

---

3. Step-by-Step Hands-On Lab

Step 1: Create a CSV Dataset with Wildcard

1. Open Azure Data Factory Studio.
2. Go to **Author** → **Datasets**.
3. Click **+ New Dataset**.
4. Select **Azure Data Lake Storage Gen2** or **Blob Storage**.
5. Choose **CSV**.
6. Name your dataset: **MultiCSVInput**.
7. Select the folder path (example: `input/`).
8. In the **file name** field, enter:

   ```
   *.csv
   ```

9. Enable **Wildcard file path**.
10. Save the dataset.

Your dataset is now ready to read multiple CSVs.

Step 2: Create a Mapping Data Flow

1. Go to **Author**.
2. Click **+** → **Data Flow**.
3. Choose **Mapping Data Flow**.
4. Name it: **LoadMultipleCSVDataFlow**.

Step 3: Add Source Transformation

1. Click **Add Source**.
2. Select the dataset **MultiCSVInput**.
3. Under **Source Options** enable:
4. **Allow schema drift**
5. **Infer drifted column types**

ADF will automatically read all CSV files.

Step 4: Add Derived Column to Capture File Name

Azure Data Factory provides a built-in function:

```
source().fileName
```

This gives the name of the current file being processed.

Steps:

1. Click **+** after the Source.
2. Select **Derived Column**.
3. Add a new column:
4. Name: `SourceFileName`
5. Expression:

   ```
   source().fileName
   ```

Now every record will have the file name added.

Step 5: Create Azure SQL Sink Dataset

1. Go to **Datasets** → **+ New Dataset**.
2. Select **Azure SQL Database**.
3. Choose or create a table like:

```sql
CREATE TABLE SalesCombined (
    Product VARCHAR(100),
    Quantity INT,
    Price DECIMAL(10,2),
    SourceFileName VARCHAR(200)
);
```

Name the dataset **SQLSalesOutput**.

Step 6: Add Sink Transformation

1. In the Data Flow, click **+** after Derived Column.

2. Select **Sink**.
3. Choose the SQL dataset: **SQLSalesOutput**.
4. In **Settings**:
5. Enable **Allow Schema Drift**.
6. Disable **Validate Schema**.
7. In **Mapping**:
8. Click **Auto Mapping**.
9. Ensure `SourceFileName` is mapped to the SQL table column.

Step 7: Debug the Data Flow

1. Turn on **Data Flow Debug**.
2. Wait until the session is ready.
3. Go to **Data Preview**.
4. Verify:
5. Data from all CSV files appears
6. `SourceFileName` column shows correct file names

Example Output: | Product | Quantity | Price | SourceFileName | |---------|----------|--------|---------------| | Pen | 10 | 20 | sales_jan.csv | | Book | 5 | 100 | sales_jan.csv | | Pen | 20 | 30 | sales_feb.csv |

Step 8: Run the Data Flow Using a Pipeline

1. Create a new pipeline.
2. Drag **Data Flow** activity to the canvas.
3. Select **LoadMultipleCSVDataFlow**.
4. Debug or Trigger the pipeline.

This will load all CSVs into the SQL table along with their file names.

---

4. Final Description

In this lab, you learned how to ingest multiple CSV files using Azure Data Factory Mapping Data Flow and load them into Azure SQL Database. You also learned how to add the file name for each record using the `source().fileName` function. This helps in tracking, auditing, and debugging your data ingestion process.

This completes the hands-on exercise.