

# Hands-On Lab: Load Multiple CSV Files to Azure SQL Database with File Names in Mapping Data Flow

**Author:** Sandeep Kumar Sharma

## Learning Objective

This lab teaches you how to load multiple CSV files from Azure Storage into an Azure SQL Database table using Mapping Data Flow in Azure Data Factory. You will also learn how to include the **source file name** as an additional column while loading the data.

## Learning Outcome

After completing this lab, you will be able to: - Read multiple CSV files using a single dataset - Access file name in Mapping Data Flow using the `source()` function - Add the file name as a new column - Load combined data into Azure SQL Database

---

## 1. Introduction

Often in data ingestion scenarios, multiple CSV files arrive daily in a folder. Each file may contain customer data, sales data, log data, etc. While loading these files into a database table, it is very important to also load the **file name** from which the data originated.

In this lab, you will: - Use wildcard paths to read multiple CSV files - Extract the file name from each file automatically - Add the file name as a new column - Insert the records into Azure SQL Database

---

## 2. Prerequisites

Make sure you have: - Azure Storage Account with a folder containing multiple CSV files - Azure SQL Database with a table created - Basic understanding of Mapping Data Flow and linked services

Example CSV files in storage:

```
/sales/2024-jan-sales.csv  
/sales/2024-feb-sales.csv  
/sales/2024-mar-sales.csv
```

---

### 3. Step-by-Step Hands-On Lab

#### Step 1: Create a Dataset for the CSV Files

1. Go to **Author** → **Datasets**.
2. Click **+ New Dataset**.
3. Select **Azure Blob Storage / ADLS Gen2**.
4. Choose **CSV format**.
5. Name it: **SalesCSVFiles**.
6. In the file path section:
  - 7. Folder Path: select your folder (e.g., **sales/**)
  - 8. File Path: leave blank or type **\*.csv**
  - 9. Enable **Wildcard file path**.
10. Save the dataset.

This will allow ADF to read all CSV files in the folder.

---

#### Step 2: Create a Mapping Data Flow

1. Click **+** → **Data Flow**.
  2. Select **Mapping Data Flow**.
  3. Name it: **LoadCSVToSQLWithFilename**.
- 

#### Step 3: Add Source Transformation

1. Click **Add Source**.
2. Choose the dataset **SalesCSVFiles**.
3. Under **Source Options**:
4. Enable **Allow Schema Drift**.
5. Enable **Infer Drifted Column Types**.

##### Step 3.1: Extract File Name

ADF provides a built-in function:

```
source().fileName
```

This gives the name of each file being processed.

---

## Step 4: Add a Derived Column to Capture File Name

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

```
source().fileName
```

This will attach the file name to every row.

---

## Step 5: Add Sink to Load Data into Azure SQL Database

1. Click + after Derived Column.
2. Select **Sink**.
3. Create or choose an **Azure SQL Database** dataset.
4. Select the table where you want to load data.

### SQL Table Example

Your table should look like:

```
CREATE TABLE SalesData (
    ProductName VARCHAR(100),
    Quantity INT,
    Amount DECIMAL(10,2),
    SourceFileName VARCHAR(200)
);
```

### Sync Settings

1. In the Sink settings:
  2. Set **Allow Schema Drift** = Yes
  3. Set **Validate Schema** = No
  4. Under **Mapping**, click **Auto Mapping** to map columns.
- 

## Step 6: Debug and Preview

1. Turn on **Data Flow Debug**.
2. Go to **Data Preview**.
3. Verify that:

4. All CSV records are combined
5. The **SourceFileName** column shows correct file names

Example: | ProductName | Quantity | Amount | SourceFileName | -----|-----|-----|-----|  
Pen | 20 | 100.00 | 2024-jan-sales.csv | Book | 10 | 50.00 | 2024-jan-sales.csv | Pen | 15 | 75.00 | 2024-feb-sales.csv |

---

## Step 7: Use Pipeline to Execute the Data Flow

1. Create a new Pipeline.
2. Drag **Data Flow** activity.
3. Select **LoadCSVToSQLWithFilename** data flow.
4. Validate the pipeline.
5. Click **Debug** or **Publish + Trigger**.

This will load all CSV files' data into the SQL table along with their file names.

---

## 4. Final Description

In this lab, you learned how to load multiple CSV files into an Azure SQL Database using Mapping Data Flow. You also learned how to capture the name of each file using the `source().fileName` function and include it as a column. This is an essential technique for building audit-ready, traceable ingestion pipelines.

This completes the hands-on exercise.