Use Case: Sales Data Processing & Transformation Using Data Flow in Azure Data Factory

Scenario:

A retail company stores daily sales data in **Azure Blob Storage (CSV format)** and wants to load it into an **Azure SQL Database** for reporting. The company requires the following transformations before loading:

1. **Filter out invalid records** (missing values in key columns).
2. **Calculate total price** (Unit Price * Quantity).
3. **Convert date format** to a standardized yyyy-MM-dd.
4. **Aggregate total sales per region**.
5. **Store cleaned & transformed data** in Azure SQL Database.

Dataset (Sales Data - CSV in Blob Storage)

File Name: daily_sales.csv

OrderID	CustomerName	Product	UnitPrice	Quantity	OrderDate	Region
1001	John Doe	Laptop	800	2	03/05/2024	West
1002	Jane Smith	Phone	500	NULL	03/06/2024	East
1003	Mike Davis	Tablet	300	5	03/05/2024	South
1004	NULL	Laptop	900	3	03/07/2024	North
1005	Sara Khan	Phone	450	4	03/06/2024	West

Step-by-Step Implementation in Azure Data Factory (ADF)

Step 1: Create Linked Services

- **Blob Storage Linked Service:** Connect to the container storing daily_sales.csv.
- **Azure SQL Database Linked Service:** Connect to the destination SQL database.

Step 2: Create Data Flow in ADF

1. Create a New Data Flow

- Go to **Azure Data Factory > Author > Data Flows**
- Click **New Data Flow** > Select **Mapping Data Flow**
- Name it **SalesDataTransformation**

---

2. Add Source (Blob Storage)

- Click **Add Source**
- Name it **SalesDataSource**
- Choose the **Linked Service** pointing to Azure Blob Storage
- Select **daily_sales.csv** as the dataset
- Configure the **Schema** by importing column names

---

3. Apply Transformations

a. Filter Out Invalid Records

- Add a **Filter** transformation after the source
- Name it **RemoveInvalidRecords**
- Use the condition:

  scss

  CopyEdit

  isNotNull(CustomerName) && isNotNull(Quantity)

  (This removes records where CustomerName or Quantity is NULL)

---

b. Calculate Total Price

- Add a **Derived Column** transformation
- Name it **CalculateTotalPrice**
- Create a new column TotalPrice with the expression:

mathematica

CopyEdit

UnitPrice * Quantity

---

c. Convert Date Format

- Add another **Derived Column** transformation
- Name it **ConvertOrderDate**
- Create a new column FormattedDate with the expression:

scss

CopyEdit

toDate(OrderDate, 'MM/dd/yyyy')

---

d. Aggregate Total Sales Per Region

- Add an **Aggregate** transformation
- Name it **TotalSalesByRegion**
- Group by Region
- Create an aggregated column TotalSales with the expression:

scss

CopyEdit

sum(TotalPrice)

---

4. Add Sink (Azure SQL Database)

- Click **Add Sink**
- Name it **SalesDataSink**
- Choose the **Azure SQL Database Linked Service**
- Select the target table **SalesReport**
- Enable **Auto Mapping** to match columns

Nexus SOLUTIONS

Step 3: Publish and Execute the Data Flow

1. Click **Publish All** to save changes.
2. Create a **Pipeline** and add the **SalesDataTransformation Data Flow**.
3. Trigger the pipeline manually or schedule execution with a **Trigger**.

Expected Output in Azure SQL Database (SalesReport Table)

Region	TotalSales
West	3800
South	1500
North	2700
East	2000

Summary of Implementation

✅ Connected **Azure Blob Storage** and **Azure SQL Database**
✅ Filtered out invalid records
✅ Computed **Total Price**
✅ Converted **Order Date Format**
✅ Aggregated **Total Sales Per Region**
✅ Loaded cleaned & transformed data into Azure SQL Database

Next Steps

- Schedule the pipeline to run **daily**.
- Enable **Monitoring & Alerts** in ADF to track failures.
- Optimize pipeline performance using **partitioning**.