**Azure Data Factory ETL Demo Lab**

**Scenario**

You are tasked with creating an **ETL pipeline** in Azure Data Factory.

* **Extract** sales data (CSV) from **Azure Blob Storage**.
* **Transform** the data by cleaning column names and aggregating total sales per product.
* **Load** the transformed data into **Azure SQL Database**.

**Step 1: Prepare Resources**

1. **Azure Blob Storage**
   * Create a container named input.
   * Upload a sample file sales.csv with the following content:
2. SaleID,Product,Quantity,Price,Date
3. 1,Laptop,2,500,2025-09-01
4. 2,Mouse,10,20,2025-09-01
5. 3,Laptop,1,500,2025-09-02
6. 4,Keyboard,5,50,2025-09-02
7. **Azure SQL Database**
   * Create a database named SalesDW.
   * Create the target table:
8. CREATE TABLE ProductSalesSummary (
9. ProductName VARCHAR(50),
10. TotalQuantity INT,
11. TotalSales DECIMAL(18,2)
12. );
13. **Azure Data Factory**
    * Create a new Data Factory instance.

**Step 2: Create Linked Services**

1. In ADF → Manage → Linked Services → Create:
   * **Blob Storage Linked Service** (use account key).
   * **Azure SQL DB Linked Service** (use SQL authentication).

**Step 3: Create Datasets**

1. **Blob Dataset**
   * Source: CSV file in input container.
   * First row has headers = True.
2. **SQL Dataset**
   * Sink: ProductSalesSummary table in SalesDW.

**Step 4: Build Pipeline**

1. Go to **Author → Pipelines → New Pipeline**.
2. Add **Data Flow Activity** instead of direct Copy, since we want transformation.

**Step 5: Design Data Flow (Transformations)**

1. Create a **Mapping Data Flow**:
   * **Source:** Blob dataset (CSV).
   * **Transformations:**
     + **Select Transformation:** Rename columns for clarity (Product → ProductName, Quantity → Qty).
     + **Derived Column Transformation:** Add new column SalesAmount = Qty \* Price.
     + **Aggregate Transformation:**
       - Group by: ProductName
       - Aggregates:
         * TotalQuantity = sum(Qty)
         * TotalSales = sum(SalesAmount)
   * **Sink:** SQL dataset (ProductSalesSummary).
2. Publish the Data Flow.

**Step 6: Trigger & Run**

1. Go back to Pipeline → Add the Data Flow activity.
2. Debug run to test execution.
3. Once validated, **Publish All**.
4. Create a **Schedule Trigger** to run pipeline daily at midnight.

**Step 7: Monitor**

1. Go to **Monitor Hub** → Check pipeline run status.
2. Verify row count and transformation logs.
3. Query SQL DB to confirm results:
4. SELECT \* FROM ProductSalesSummary;

Expected Output:

| **ProductName** | **TotalQuantity** | **TotalSales** |
| --- | --- | --- |
| Laptop | 3 | 1500.00 |
| Mouse | 10 | 200.00 |
| Keyboard | 5 | 250.00 |

**Step 8: Error Handling**

1. In Data Flow activity settings → configure:
   * **Retry = 2**
   * **Timeout = 30 min**
2. Upload a malformed CSV (missing a column).
3. Run pipeline → Observe failure logs in Monitor.

**Step 9: Cost Awareness**

1. Open **Monitor → Integration Runtime metrics** → check data flow compute usage.
2. Use **Azure Pricing Calculator** to estimate cost for daily pipeline runs.