Azure Data Factory Project Report

1. Project Overview

This project demonstrates end-to-end data integration using Azure Data Factory (ADF). It extracts data from an on-premise MySQL server and loads it into an Azure SQL Database. It uses a Self-hosted Integration Runtime (SHIR), incremental load techniques, and scheduled triggers.

2. Setting up the Azure Environment

- Created a Resource Group in Azure.

- Set up Azure Data Factory instance under the Resource Group.

- Created an Azure SQL Database and noted credentials.

3. Installing and Configuring SHIR

- Installed Self-hosted Integration Runtime (SHIR) on local system.

- Registered it in ADF using the generated authentication key.

- Confirmed SHIR status as 'Running' in ADF Studio.

4. Creating Linked Services

- Created a Linked Service to connect to the on-premise MySQL using SHIR.

- Created another Linked Service to connect to Azure SQL Database.

- Tested both connections successfully.

5. Creating Datasets

- Dataset for MySQL source table 'Employees' was defined using the linked service.

- Dataset for Azure SQL sink table was created similarly.

6. Creating the Pipeline

- A pipeline was created using ?Copy Data? activity.

- Source tab used a query with a dynamic parameter for incremental loading based on 'LastModified'.

- Sink tab mapped the schema appropriately.

7. Setting Parameters and Triggers

- Created pipeline parameter 'LastModifiedDate'.

- Configured a trigger to run the pipeline daily with a default watermark value.

- Added a custom trigger using CRON expression to run on the last Saturday of every month.

8. Testing and Validation

- Used 'Validate All' to ensure pipeline had no issues.

- Screenshots were taken at every stage to document completion.

- Debug runs were triggered where supported.