End-to-End ETL Project Report

# 1. Project Overview

This project demonstrates building a scalable ETL pipeline from on-premise data sources to Azure SQL Database using a modern Azure data architecture.   
The goal is to extract raw data from multiple sources, validate and transform it in Databricks, and load it into an integration layer with Slowly Changing Dimensions (SCD) logic for analytical use.

# 2. Architecture Flow

Step-by-step flow:  
1. Source Data  
 - 6 CSV files in on-premise file system.  
 - 6 relational tables in on-premise SQL Server.  
  
2. Landing Zone (Blob Storage)  
 - Using ADF Copy Activity, moved both CSV and SQL tables into Azure Blob Storage (raw zone).  
  
3. Data Lake (ADLS Gen2)  
 - From Blob, ADF pipelines copy data into ADLS Gen2 for structured storage.  
  
4. Validation & Processing (Databricks)  
 - Databricks reads all ADLS data, performs schema validation, null checks, duplicate removal.  
 - After validation, writes clean data to Azure SQL Database (Staging Layer).  
  
5. Integration Layer (Azure SQL Database)  
 - Applied SCD1 and SCD2 logic using SQL MERGE.  
 - Final dimension tables include TBL\_DIM\_ORDER, TBL\_DIM\_PRODUCT, and multiple lookup dimensions.  
  
6. Orchestration (ADF Pipelines)  
 - Master Pipeline (daily at 9 AM UTC).  
 - Ad-hoc Pipeline (lookup loads on-demand).  
 - Failure Handling: Sp\_Log\_Failure procedure for error logging.

# 3. Technology Stack

- Azure Data Factory (ADF) → Data movement & orchestration.  
- Azure Blob Storage → Initial landing zone.  
- ADLS Gen2 → Raw + curated data lake.  
- Azure Databricks → Data validation and transformation.  
- Azure SQL Database → Staging + Integration layer with SCD logic.  
- On-Prem SQL Server & CSV → Source systems.  
- GitHub → Version control for pipelines and SQL scripts.

# 4. Data Model

Dimension Tables:  
- TBL\_DIM\_ORDER (SCD2) – keeps history of order attributes.  
- TBL\_DIM\_PRODUCT (SCD1) – updates in place for product details.  
- Lookup Tables (SCD1, Ad-hoc loads):  
 - TBL\_DIM\_ORDER\_METHOD\_LKP  
 - TBL\_DIM\_RETURN\_REASON\_LKP  
 - TBL\_DIM\_COUNTRY\_LKP  
 - TBL\_DIM\_WAREHOUSE\_LKP  
 - TBL\_DIM\_RETAILER\_LKP  
 - TBL\_DIM\_PRODUCT\_NAME\_LKP

# 5. ETL Logic

SCD1-

MERGE Integration.TBL\_DIM\_PRODUCT\_NAME\_LKP AS tgt

USING (

SELECT PRODUCT\_NUMBER, PRODUCT\_NAME,PRODUCT\_DESCRIPTION

FROM stg.PRODUCT\_NAME\_LOOKUP

) AS src

ON tgt.PRODUCT\_NUMBER = src.PRODUCT\_NUMBER

WHEN MATCHED THEN

UPDATE SET tgt.PRODUCT\_NAME = src.PRODUCT\_NAME,

tgt.PRODUCT\_DESCRIPTION=src.PRODUCT\_DESCRIPTION,

tgt.UPDATE\_DATE = GETDATE()

-- If record is new, insert it

WHEN NOT MATCHED THEN

INSERT (PRODUCT\_NUMBER, PRODUCT\_NAME,PRODUCT\_DESCRIPTION, SOURCE\_ID, DATA\_DATE, UPDATE\_DATE)

VALUES (src.PRODUCT\_NUMBER, src.PRODUCT\_NAME,src.PRODUCT\_DESCRIPTION, 'LKP\_FILE',GETDATE() , GETDATE());

# 6. ADF Pipeline Design

Master Pipeline:  
- Executes child pipelines (Order, Product, etc.).  
- Runs Stored Procedure Activity (SCD logic).  
- Failure path → Sp\_Log\_Failure.  
  
Ad-hoc Pipeline:  
- Only Lookup tables, run on-demand.  
- Trigger type: Manual.

# 7. Failure Logging

ETL\_Log Table:  
CREATE TABLE ETL\_Log (  
 LogId INT IDENTITY PRIMARY KEY,  
 PipelineName NVARCHAR(100),  
 ActivityName NVARCHAR(100),  
 TableName NVARCHAR(100),  
 ErrorMessage NVARCHAR(MAX),  
 RunDate DATETIME DEFAULT GETDATE()  
);  
  
Sp\_Log\_Failure procedure inserts records into ETL\_Log.  
ADF calls this proc via failure path.

# 8. Scheduling & Triggers

- Daily Trigger: Master pipeline at 9 AM UTC.  
- Ad-hoc Trigger: Manual execution for lookup tables.

# 9. Monitoring

- ADF Monitoring dashboard shows pipeline runs, success/failure, logs.  
- SQL ETL\_Log table records error details for audit.

# 10. GitHub Repository Structure

/sql\_scripts  
 create\_tables.sql  
 store\_procedure.sql  
   
  
/adf\_pipelines  
 MasterPipeline.json  
 AdhocPipeline.json  
 ChildPipelines.json  
  
/docs  
 Project\_Report.md  
 Architecture.png  
 Pipelines.png  
  
README.md