REQUIREMENTS DOCUMENT

DATA WAREHOUSING INTEGRATION

1. INTRODUCTION

1.1 PURPOSE

The purpose of this project is to design and implement a **Data Warehousing Integration (DWI) solution** using Azure cloud services. The solution enables automated ingestion, transformation, and loading (ETL) of e-commerce datasets into a centralized warehouse, making data accessible for analytics and reporting.

1.2 BACKGROUND

In the retail/e-commerce industry, huge volumes of data are generated daily from customer interactions, product catalog updates, and sales transactions. Without a proper data warehouse, data is fragmented across different systems, making it difficult to answer key business questions such as:

- Which products are selling the most?
- Who are the top customers by revenue?
- What are the revenue trends over time?

A **data warehouse** addresses these issues by consolidating data into a single source of truth. Using Azure Data Factory (ADF), Azure Databricks, and Azure Synapse Analytics, we can build an automated pipeline that processes raw data and makes it ready for analysis.

1.3 SCOPE

1.3.1 In-Scope:

- Ingestion of CSV datasets into Azure Data Lake Storage (ADLS) using Azure Data Factory.

- Transformation of raw data into clean, structured formats using Azure Databricks with PySpark.
- Loading curated data into Azure Synapse Analytics in a star schema format.
- Daily pipeline scheduling and monitoring using ADF triggers.
- Validation queries to ensure data accuracy.
- **Optional**: Connection of Power BI to Synapse for visualization.

1.3.2 Out-of-Scope:

- Real-time or streaming ingestion.
- Advanced machine learning and AI analytics.
- External third-party API integrations.

2. BUSINESS REQUIREMENTS

2.1 BUSINESS OBJECTIVES

- Consolidate fragmented data into a centralized warehouse.
- Automate daily ETL processes to reduce manual intervention.
- Provide reliable and timely data for analytics.

2.2 EXPECTED BUSINESS OUTCOMES

- Improved decision-making through accurate reporting.
- Faster access to curated datasets for analysts.
- Reduction in operational inefficiencies caused by manual data handling.

2.3 SUCCESS CRITERIA

- ≥ 99% pipeline success rate.
- End-to-end processing within 15 minutes for 1M+ records.
- Data available in Synapse by **T+1 (next business day)**.

3. TECHNICAL REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS (FR)

The functional requirements describe **what the system should do**. These are critical for ensuring the solution performs its intended purpose.

FR ID	Requirement	Description
FR1	Ingest data into ADLS	ADF must ingest CSV files into a staging container in ADLS.
FR2	Transform raw data	Databricks (PySpark) must clean nulls, remove duplicates, format dates, and calculate total sales (Quantity * Price).
FR3	Join datasets	Orders must be enriched with customer and product details to create a FactSales dataset.
FR4	Load into Synapse	Transformed data must be loaded into Synapse Analytics into DimCustomer, DimProduct, DimDate, and FactSales tables.
FR5	Schedule runs	Pipelines must be scheduled to run daily at a fixed time (T+1).
FR6	Monitor pipelines	ADF must log execution results and notify on failures.

3.2 NON-FUNCTIONAL REQUIREMENTS (NFR)

Non-functional requirements define the **quality attributes** of the system such as performance, scalability, security, and cost control.

NFR ID	Requirement	Description
NFR1	Performance	The pipeline must process 1M+ order records within 15 minutes.
NFR2	Scalability	Databricks must support autoscaling clusters to handle data growth without manual intervention.
NFR3	Security	Use SAS tokens/Key Vault, enforce role-based access, no public exposure.
NFR4	Reliability	Pipelines must retry on failure up to 3 times and be idempotent (can re-run without data duplication).
NFR5	Cost Optimization	Databricks job clusters with auto-termination must be used to minimize costs.

3.3 REQUIREMENT PRIORITIZATION

The requirements are prioritized into **High**, **Medium**, and **Low**, based on their impact on project success.

Priority	Requirements
High	FR1 (Ingestion), FR2 (Transformation), FR3 (Joins), FR4 (Loading), NFR1 (Performance), NFR3 (Security)
Medium	FR5 (Scheduling), FR6 (Monitoring), NFR2 (Scalability), NFR4 (Reliability)
Low	NFR5 (Cost optimization), Power BI dashboards (optional)

4. DELIVERABLES

- **Data Pipelines (ADF):** Orchestrated workflows for raw → curated → Synapse.
- **Databricks Notebooks:** Transformation logic in PySpark.
- **Data Warehouse Schema:** Star schema with FactSales + Dimension tables.

- **Validation Queries:** SQL checks for data consistency (row counts, orphaned keys, trends).

- **Monitoring & Scheduling:** Daily triggers, run logs, error handling.

5. USE CASE OVERVIEW

5.1 ACTORS

- **Data Engineer** – Designs and manages pipelines.

- **System (ADF + Databricks + Synapse)** – Executes ingestion, transformation, and loading automatically.

- **Analyst/Business User** – Queries the Synapse warehouse for insights.

5.2 USE CASES

1. **Ingest Data**: ADF copies raw CSV files from source into ADLS staging.

2. **Transform Data**: Databricks processes the raw data, cleaning and enriching it.

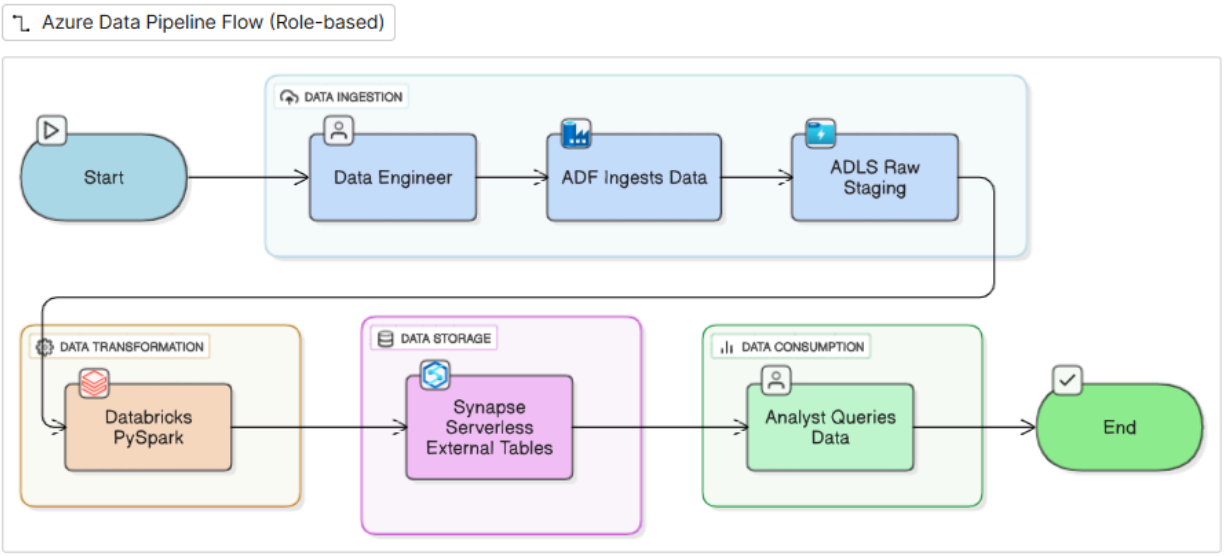
3. **Load Data**: The processed data is written into Synapse external tables.

4. **Analyze Data**: Analysts query the data in Synapse to generate reports (e.g., total sales by region).

5. **Visualize Insights:** Power BI connects to Synapse for interactive dashboards and business reporting.

5.3 DIAGRAM (CONCEPTUAL FLOW)

This use case flow shows how data travels from source to warehouse, and finally to the end-user.



6. DATA MODEL

6.1 ENTITIES

- **Customer:** CustomerID, Gender, Age, Region
- **Product:** ProductID, Category, Price
- **Sales/Orders:** TransactionID, CustomerID, ProductID, Date, Quantity, TotalAmount
- **Date:** DateID, Year, Month, Day, Quarter

6.2 WAREHOUSE TABLES

- **DimCustomer:** Surrogate key, CustomerID, Gender, Age
- **DimProduct:** Surrogate key, ProductID, Category
- **DimDate:** Surrogate key, DateID, Date, Year, Month, Day, Quarter
- **FactSales:** TransactionID, CustomerKey, ProductKey, DateKey, Quantity, Price, TotalAmount

6.3 RELATIONSHIPS

- One Customer → Many Sales
- One Product → Many Sales
- FactSales links DimCustomer, DimProduct, and DimDate

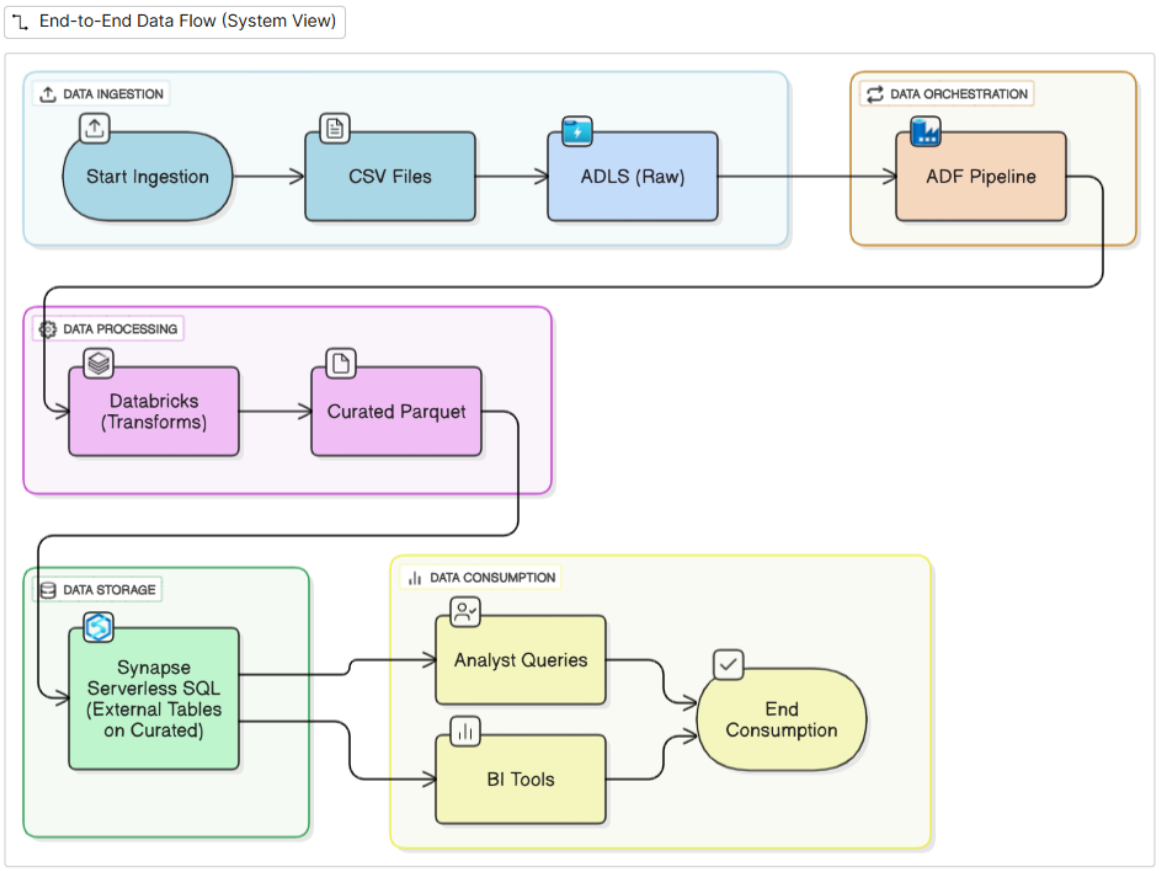
7. ADDITIONAL ENHANCEMENTS

7.1 HIGH-LEVEL ARCHITECTURE

- **ADF:** Orchestrates ingestion and scheduling.
- **Databricks:** Performs transformations with PySpark.
- **ADLS:** Stores raw and curated datasets.
- **Synapse:** Hosts external tables for analytics.

7.2 ARCHITECTURE DIAGRAM

A simple block flow:



8. ACCEPTANCE CRITERIA

- Pipelines complete successfully with ≥ 99% reliability.
- End-to-end execution within 15 minutes for large datasets.
- Curated data accessible in Synapse by **next day (T+1)**.
- Validation queries confirm consistency (row counts, joins, sales summaries).

9. RISKS & MITIGATION

S.No.	Risk	Mitigation
1	Schema drift	Use Databricks schema evolution to adapt to new columns.
2	Data quality issues	Apply null/duplicate checks in transformations.
3	Cost overruns	Enforce job cluster auto-termination and budget alerts.
4	Dashboard performance issues	Optimize Synapse queries and Power BI data models.
5	Credential mismanagement	Use Key Vault / SAS tokens with strict expiry.

10. STAKEHOLDERS & ROLES

- **Data Engineer** – Implements ADF pipelines and transformations.
- **System (ADF + Databricks + Synapse)** – Automates workflow execution.
- **Analyst/Business User** – Queries Synapse tables to derive insights.
