**DATA WAREHOUSING INTEGRATION**

[*Name : Vijaya Harcini J* ]

[*Mail Id : kit.25.21bad061@gmail.com*]

**TABLE OF CONTENTS**

**1. Introduction** 1.1 Purpose  
 1.2 Background  
 1.3 Scope

**2. Problem Statement**

**3. Objectives**

**4. Tools & Technologies Used** 4.1 Azure Data Factory (ADF)  
 4.2 Azure Data Lake Storage (ADLS)  
 4.3 Azure Databricks (PySpark)  
 4.4 Azure Synapse Analytics

**5. Solution Architecture** 5.1 High-Level Design  
 5.2 Data Flow Diagram  
 5.3 Entity Relationship Diagram (ERD)

**6. Execution Overview** 6.1 Environment Setup  
 6.2 Data Ingestion  
 6.3 Data Transformation  
 6.4 Data Loading  
 6.5 Validation & Testing  
 6.6 Orchestration & Monitoring

**7. Tasks Performed** 7.1 Environment Setup Tasks  
 7.2 Data Ingestion Tasks  
 7.3 Transformation Tasks  
 7.4 Data Loading Tasks  
 7.5 Validation Queries & Results  
 7.6 Pipeline Orchestration Tasks

**8. Results & Insights** 8.1 Row Counts  
 8.2 Referential Integrity Checks  
 8.3 Aggregated Insights (Product, Customer, Gender, Time-based trends)

**9. Key Learnings**

**10. Challenges & Solutions** 10.1 SAS Token Issues  
 10.2 External Table Accessibility Errors  
 10.3 Credential & Data Source Dependencies  
 10.4 Schema Mismatches

**11. Deliverables** 11.1 Requirement Document  
 11.2 Project Overview Document  
 11.3 Execution Overview Document  
 11.4 Results & Insights Document  
 11.5 Final Merged Project Document  
 11.6 PowerPoint Presentation

**12. Future Scope**

**13. Conclusion**

## **1. INTRODUCTION**

### **1.1 PURPOSE**

The purpose of this project is to design and implement a data warehousing integration solution using Microsoft Azure services. The solution automates data ingestion, transformation, and loading (ETL) from raw sources into a centralized warehouse, enabling scalable and reliable analytics for business users.

### **1.2 BACKGROUND**

In the retail and e-commerce industry, large amounts of data are generated every day from transactions, customers, and product interactions. Without a consolidated data warehouse, this data remains fragmented, making it difficult to answer key business questions such as sales performance, customer behavior, and product profitability. By leveraging Azure services, this project addresses these gaps with a modern cloud-based data warehouse.

### **1.3 SCOPE**

* **In-Scope:** Ingesting raw CSV files, transforming datasets into curated formats, creating star schema models in Synapse, and validating data through SQL queries.
* **Out-of-Scope:** Real-time streaming pipelines, advanced machine learning, and external API integrations.

## **2. PROBLEM STATEMENT**

Organizations often struggle to consolidate fragmented datasets across multiple systems, leading to reporting delays, inconsistent metrics, and poor decision-making. This project addresses the problem by building an automated ETL pipeline that ensures clean, structured, and analytics-ready data is always available in Azure Synapse.

## **3. OBJECTIVES**

* Automate data ingestion from source to data lake.
* Transform data into structured fact and dimension tables.
* Load curated datasets into Synapse for analytics.
* Validate results to ensure data quality and consistency.
* Establish a scalable pipeline framework for future BI reporting.

## **4. TOOLS & TECHNOLOGIES USED**

### **4.1 AZURE DATA FACTORY (ADF)**

Used for orchestrating data pipelines, managing linked services, and scheduling ETL processes.

### **4.2 AZURE DATA LAKE STORAGE (ADLS)**

Serves as the centralized data repository with structured zones: *raw* for ingestion and *curated* for transformed outputs.

### **4.3 AZURE DATABRICKS (PYSPARK)**

Provides scalable data transformation capabilities, enabling cleaning, enrichment, and preparation of datasets for Synapse.

### **4.4 AZURE SYNAPSE ANALYTICS**

Acts as the data warehouse, hosting fact and dimension tables through external table mappings to curated data.

## **5. SOLUTION ARCHITECTURE**

### **5.1 HIGH-LEVEL DESIGN**

The architecture follows a modular ETL approach: data is ingested by ADF, stored in ADLS, transformed in Databricks, and loaded into Synapse for reporting.

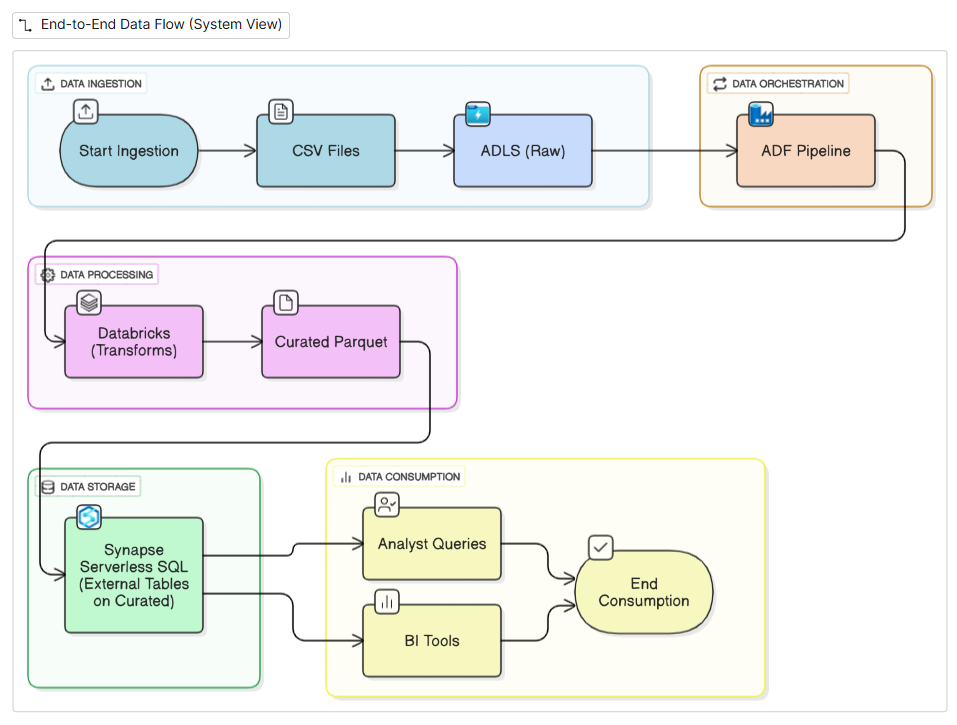
### 

### 

### 

### 

### **5.2 DATA FLOW DIAGRAM**



### **5.3 ENTITY RELATIONSHIP DIAGRAM (ERD)**Generated image

* **DimCustomer** (CustomerKey, CustomerID, Name, Gender, Age).
* **DimProduct** (ProductKey, ProductID, ProductName, Category, Price).
* **DimDate** (DateKey, Date, Year, Month, Day).
* **FactSales** (SalesKey, CustomerKey, ProductKey, DateKey, Quantity, TotalAmount).

## 

## **6. EXECUTION OVERVIEW**

### **6.1 ENVIRONMENT SETUP**

Provisioned ADLS, ADF, Databricks, and Synapse workspaces. Created storage containers (raw/, curated/). Configured linked services and secured access with SAS tokens.

### **6.2 DATA INGESTION**

Implemented ADF pipelines to copy raw CSVs into ADLS staging. Handled credential issues by resetting admin details and correcting Synapse server configuration.

### **6.3 DATA TRANSFORMATION**

Developed PySpark scripts in Databricks to clean nulls, remove duplicates, calculate sales metrics, and create fact/dimension tables. Outputs were stored in Parquet format under /curated/.

### **6.4 DATA LOADING**

Created scoped credentials and external data sources in Synapse. Registered external tables (dim\_customer, dim\_product, dim\_date, fact\_sales) mapped to curated Parquet files.

### **6.5 VALIDATION & TESTING**

Ran SQL queries to validate row counts, integrity between fact and dimension tables, and perform aggregations (e.g., sales by product, customer, gender, and month).

### **6.6 ORCHESTRATION & MONITORING**

Configured end-to-end orchestration in ADF. Pipelines executed in sequence (Ingest → Transform → Load → Validate). Monitoring dashboards confirmed successful runs.

## **7. TASKS PERFORMED**

## **7.1 ENVIRONMENT SETUP TASKS**

* Created ADLS with hierarchical namespace.
* Provisioned Databricks workspace and linked it to ADLS.
* Set up Synapse workspace and SQL pools.

### **7.2 DATA INGESTION TASKS**

* Configured ADF linked services for ADLS and Synapse.
* Built pipelines to move raw CSVs into staging.

### **7.3 TRANSFORMATION TASKS**

* Implemented PySpark jobs for data cleaning and enrichment.
* Generated curated fact and dimension tables.

### **7.4 DATA LOADING TASKS**

* Registered external data sources and credentials in Synapse.
* Created external tables referencing curated Parquet data.

### **7.5 VALIDATION QUERIES & RESULTS**

* Validated row counts in Synapse.
* Confirmed fact-dimension relationships.
* Verified aggregations matched curated outputs.

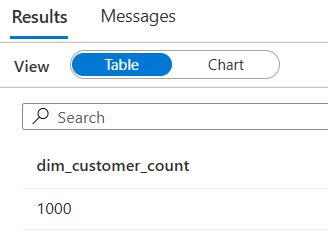
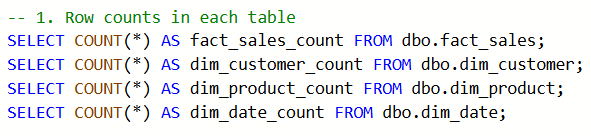
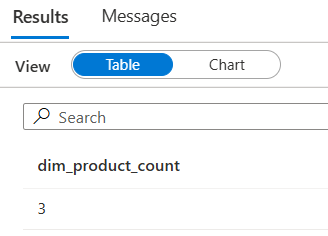
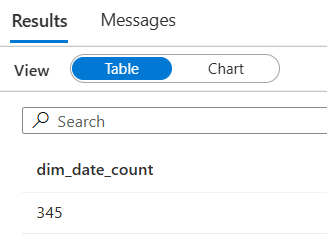
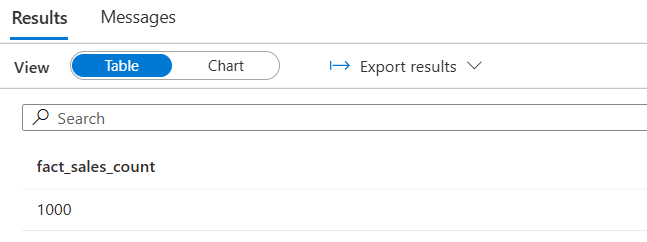
### **7.6 PIPELINE ORCHESTRATION TASKS**

* Designed ADF workflows with retry logic.
* Debugged errors in linked services and external tables.

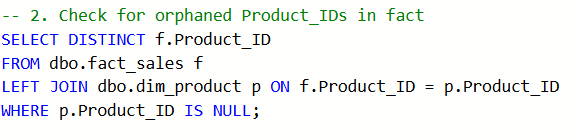
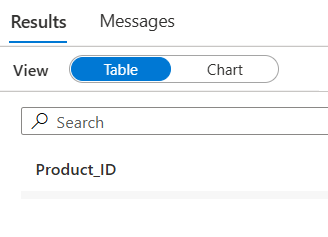
## **8. RESULTS & INSIGHTS**

### **8.1 ROW COUNTS**

Validated successful ingestion with expected row counts across fact and dimension tables.



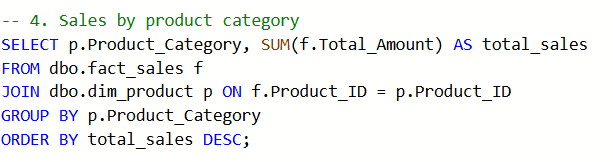
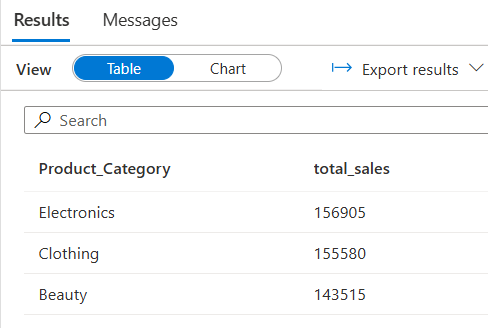
### **8.2 REFERENTIAL INTEGRITY CHECKS**

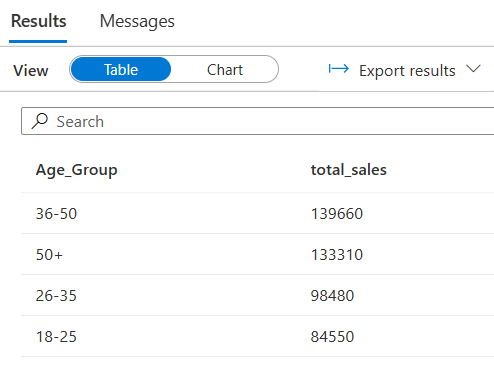
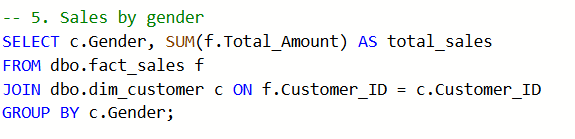
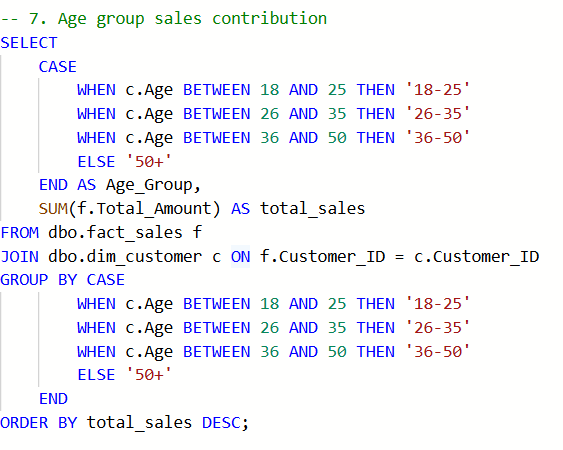
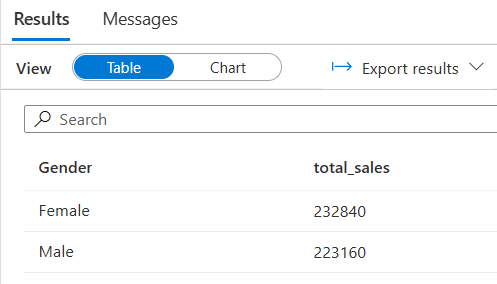
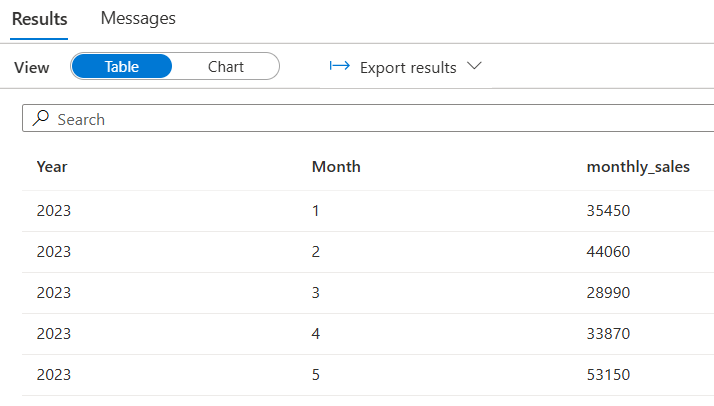
Confirmed no orphaned records existed between fact and dimension tables.

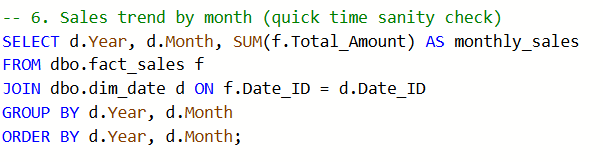
### 

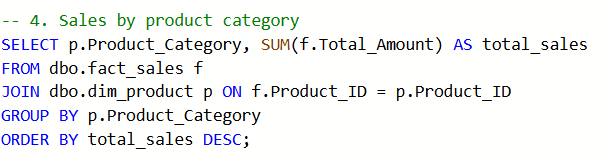
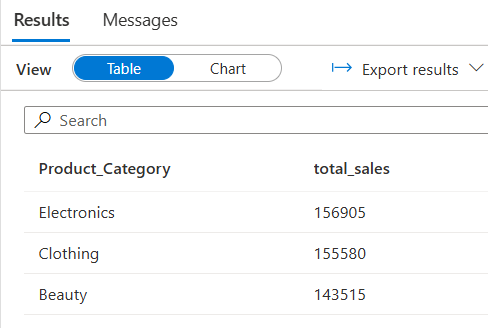
### **8.3 AGGREGATED INSIGHTS**

* Top-selling products identified.



* Sales trends by gender and age groups.
* Revenue patterns by month and product category.





## **9. KEY LEARNINGS**

* SAS token scoping must be carefully managed.
* Iterative debugging improves reliability of external tables.
* Modular ETL design improves scalability and maintainability.

## **10. CHALLENGES & SOLUTIONS**

### **10.1 SAS TOKEN ISSUES**

Resolved by regenerating Blob-only SAS tokens with proper permissions.

### **10.2 EXTERNAL TABLE ACCESSIBILITY ERRORS**

Fixed by remapping credentials and data sources in Synapse.

### 

### **10.3 CREDENTIAL & DATA SOURCE DEPENDENCIES**

Addressed by removing existing dependencies before recreating credentials.

### **10.4 SCHEMA MISMATCHES**

Resolved by aligning data types and ensuring schema consistency across transformations.

## **11. DELIVERABLES**

* Requirement Document
* Project Overview Document
* Execution Overview Document
* Results & Insights Document
* Final Project Document
* PowerPoint Presentation

## **12. FUTURE SCOPE**

* Enable real-time ingestion pipelines.
* Integrate Power BI dashboards for business reporting.
* Extend the warehouse to support machine learning workloads.

## **13. CONCLUSION**

The project successfully implemented an automated data warehousing pipeline integrating ADF, Databricks, ADLS, and Synapse. With curated datasets structured into a star schema, the solution provides reliable, scalable, and secure foundations for analytics. This project not only met its objectives but also laid the groundwork for future BI and advanced analytics initiatives.