EXECUTION OVERVIEW DOCUMENT

DATA WAREHOUSING INTEGRATION

1. EXECUTION OVERVIEW

This project was executed in a structured and iterative manner to ensure smooth integration of Azure services for end-to-end data warehousing. The objective was to design a reliable ETL pipeline that ingests raw e-commerce data, transforms it into curated datasets, and loads it into a star schema inside Azure Synapse Analytics for analytical consumption.

The execution involved the following phases:

- **Environment Setup:** Provisioned and configured Azure resources including Azure Data Lake Storage (ADLS), Azure Data Factory (ADF), Azure Databricks, and Azure Synapse Analytics.

- **Data Ingestion:** Implemented ADF pipelines to move raw CSV files (Customers, Products, Sales Transactions) into ADLS staging zones.

- **Data Transformation:** Leveraged Databricks notebooks with PySpark for large-scale cleaning, normalization, and enrichment of raw datasets, ensuring they were analytics-ready.

- **Data Loading:** Registered curated datasets as external tables in Synapse, with correct SAS credentialing, external data source mapping, and schema consistency.

- **Validation & Testing:** Executed validation queries in Synapse to confirm data quality, schema alignment, and referential integrity.

- **Pipeline Orchestration:** Automated the entire ETL process using ADF pipelines with monitoring, retries, and error handling.

Throughout execution, multiple technical challenges were encountered and resolved — including SAS token misconfigurations, Synapse external table access issues, and linked service credential dependencies. By carefully redesigning the SAS scope and repeatedly testing pipeline execution, stable connectivity and reliable data queryability were achieved.

2. TASKS PERFORMED

Below is the expanded step-by-step log of the tasks carried out during execution.

Step 1: Environment Setup

- Created an Azure Storage Account with Hierarchical Namespace (Data Lake Gen2) enabled. This provided the central data lake for raw and curated datasets.
- Established a structured folder hierarchy within ADLS for clarity:
    - `/raw/` → landing zone for unprocessed source files.
    - `/curated/` → cleaned, transformed data ready for analytics.
- Deployed an Azure Synapse Analytics serverless pool (logical warehouse for queries) and ensured SQL Admin credentials were configured.
- Configured Azure Databricks workspace, integrated with ADLS using Service Principal credentials and cluster permissions. This was the backbone for scalable transformations.

Step 2: Data Ingestion with ADF

- Configured Linked Services in ADF:
    - ADLS → to store raw and curated data.
    - Synapse → for external table registration and query execution.
- Developed ADF pipelines:
    - `Copy_Raw_To_curated`: Responsible for moving CSVs into ADLS `/raw/`. This provided a repeatable and monitored ingestion mechanism.
    - `Run_ETL_Notebook`: Orchestrated Databricks notebooks for transformations. It ensured ingestion and transformations were connected seamlessly.
    - `Register_Synapse_External_Tables`: Executed T-SQL scripts to create external tables in Synapse pointing to curated data.

Step 3: Data Transformation in Databricks (PySpark)

- Implemented PySpark scripts that performed:
    - Data cleaning: Removing null values, duplicate rows, and inconsistent records.
    - Standardization: Ensuring dates, numerical fields, and categorical data were in consistent formats.
    - Enrichment: Derived new columns such as `Total_Amount = Quantity * Price_per_Unit`.
- Modeled data into warehouse-ready structures:
    - Dimension Tables:
        - `dim_customer` → customer profiles with IDs, gender, age.
        - `dim_product` → product categories and identifiers.
        - `dim_date` → date-related breakdown (year, month, day, quarter, etc.).
    - Fact Table:
        - `fact_sales` → core transactional data referencing the dimensions.
- Wrote transformed outputs back into `/curated/` in Parquet format, ensuring efficient query performance in Synapse.

Step 4: Data Loading into Synapse

- Created Database Master Key and Database Scoped Credential in Synapse for secure ADLS access.
- Registered an External Data Source pointing to the ADLS curated container.
- Created External File Format (`ParquetFileFormat`) to define how Synapse interprets curated Parquet files.

- Registered external tables:
    - `dbo.dim_customer`
    - `dbo.dim_product`
    - `dbo.dim_date`
    - `dbo.fact_sales`

Step 5: Validation Queries in Synapse

- Ran validation queries to ensure correctness and completeness:
    - Row counts: Verified record counts across fact and dimension tables.
    - Referential Integrity: Checked that all foreign keys in `fact_sales` mapped to valid dimension keys (no orphaned IDs).
    - Aggregations:
        - Sales by product category.
        - Sales by gender.
        - Monthly and yearly sales trends.
        - Contribution of different age groups to total sales.
        - Top 10 customers by total spend.
- These queries confirmed the curated data aligned with the designed star schema and business expectations.

Step 6: Orchestration & Monitoring (ADF)

- Finalized the end-to-end pipeline in ADF:
    - Ingest Raw Data → Copy from source to ADLS raw zone.
    - Transform Data → Databricks notebook cleans and processes data to curated zone.
    - Register Tables in Synapse → Create/refresh external tables.
    - Validate Outputs → Query to confirm counts and integrity.
- Configured monitoring features:
    - Enabled pipeline run history and activity logs in ADF.
    - Implemented error handling & retries to ensure reliability.
- Successfully published all pipelines, ensuring they can run on schedule or be triggered manually for ad-hoc refreshes.

3. CHALLENGES & RESOLUTIONS

S.No.	Challenge	Resolution
1	Synapse connection errors – Linked service could not authenticate using default server name.	Identified that Synapse serverless endpoints require the `-ondemand` suffix. Updated linked service configuration and reset SQL admin password.
2	SQL script execution failure (`GO` keyword issue)	Removed unsupported `GO` statements in Synapse SQL script, re-ran successfully.
3	External tables not accessible (“directory cannot be listed”)	Regenerated SAS token with correct Blob service + Read/List permissions

		only. Updated both ADF and Synapse credentials.
4	Credential lock (unable to drop) – BlobSAS credential could not be dropped as it was tied to an active data source.	Dropped External Data Source first, then re-created both Data Source and Credential with a new SAS token.
5	Schema mismatches – Product_ID datatype mismatch caused fact-product join errors.	Standardized schema in Databricks transformations and redefined external tables with consistent data types.

4. SUMMARY OF EXECUTION

- The project successfully implemented a robust ETL pipeline using Azure services.
- Data now flows seamlessly from:
  Raw ingestion → Curated transformation → Synapse star schema → Validation queries → BI readiness.
- All functional requirements (FR1–FR6) and non-functional requirements (NFR1–NFR5) were fulfilled.
- The system is scalable, secure, and reliable, and fully ready for integration with BI tools like Power BI for dashboards and analytics.
