**Microsoft Fabric Project Documentation- Mohammeduzair Sayyed**

**Project Title: Enterprise Data Warehouse Implementation using Microsoft Fabric (Medallion Architecture)**

**Objective:** Design and implement a robust end-to-end data pipeline using Microsoft Fabric based on the Medallion Architecture, including Bronze, Silver, and Gold layers, with integrated reporting through Power BI.

**1. Project Architecture (Medallion Layers)**

**Bronze Lakehouse:**

* Raw data ingestion from ADLS Gen 2(Product.csv, Region.csv, Sales.csv).
* Used pipelines to copy each file into Lakehouse Bronze.
* Logging implemented to track file name, ingestion time, row count.

**Silver Lakehouse:**

* Data transformed and converted into Delta Tables using Notebooks.
* File-specific transformations applied.
* Created Delta tables:
  + product\_delta
  + region\_delta
  + sales\_delta

**Gold Warehouse (Goal initially):**

* Intended for final tables, aggregations, views, and fact generation.
* SQL Scripts created:
  + Table creation scripts for product, region, sales.

**Issue:**

* Warehouse copy operations failed due to type mismatch (e.g., Parquet UTF8 vs FLOAT).
* COPY INTO failed, leading to alternate approach using Notebooks.
* Data Warehouse generation failed.
* SQL endpoint generation failed in lakehouse, couldn’t connect to PowerBI

**2. Key Learnings and Challenges**

**Interesting Findings:**

* Warehouse scripting in Fabric requires each CREATE VIEW or PROCEDURE to be the first statement in a batch.
* COPY INTO in Warehouse has strict data type compatibility issues with Parquet files.(UTF-8000) issues.
* Connecting Power BI via Warehouse or Lakehouse (Direct Lake) is seamless if SQL endpoints are generated
* Fabric Notebooks can’t directly connect to Warehouse — only to Lakehouses.

**Difficulties Faced:**

* COPY INTO Warehouse failing due to FLOAT column being read as UTF8 BYTE\_ARRAY.
* Warehouse table creation syntax needed to be precise (no IF NOT EXISTS).
* SQL views and procedures had strict syntactic placement issues.
* Mismatch between expected formats in Warehouse and actual Delta storage in Silver layer.

**3. Final Approach**

**Data Flow:**

1. **Ingest from ADLS Gen 2to Bronze** using Fabric Pipelines.
2. **Transform in Silver** using Notebooks (Delta format).
3. **Aggregate and Join in Notebook (PySpark + Spark SQL)** using data from Silver.
4. **Write back aggregated data to Gold Lakehouse as Delta Tables**.

**Final Notebook Tasks:**

* Join sales, product, region tables.
* Perform aggregations:
* Save results to Gold Lakehouse Delta Tables.

**4. Power BI Integration**

* Tried to connect to Gold Lakehouse via Direct Lake mode.
* Build reports using aggregated Delta tables.

**5. Files and Assets Created**

| **Asset Type** | **Name** | **Layer** |
| --- | --- | --- |
| Pipeline | ingest\_product\_to\_bronze | Bronze |
| Pipeline | ingest\_region\_to\_bronze | Bronze |
| Pipeline | ingest\_sales\_to\_bronze | Bronze |
| Delta Table | product\_delta | Silver |
| Delta Table | region\_delta | Silver |
| Delta Table | sales\_delta | Silver |
| SQL Table | product | Gold |
| SQL Table | region | Gold |
| SQL Table | sales | Gold |
| Notebook | silver\_to\_gold\_aggregator | Silver input / Gold output |

**Status:**

* Silver layer completed.
* Aggregations finalized.
* Gold tables created via Notebook.
* Final copy to Gold Lakehouse successful.
* Power BI connection ready.

**Next Steps:**

* Final visualizations in Power BI.