**Project 1: Azure Data Migration and Ingestion Pipeline for Northeastern University with Microsoft Fabric**

**Objective:**

Migrate and ingest student and financial data from on-premise SQL Server databases to Azure, ensuring seamless data transition, governance, and actionable insights through Power BI and Microsoft Fabric.

**Step-by-Step Implementation:**

**Step 1: Create Azure Subscription**

* **Action:** I set up an Azure subscription for Northeastern University.
  + **I p**rovisioned an Azure subscription, ensuring necessary permissions and budgets were allocated.

**Step 2: Data Migration**

**Source Data:**

* **Database:** On-premise SQL Server
* **Tables:**
  + Students (Columns: StudentID, FirstName, LastName, DOB, Email, GPA)
  + Finances (Columns: TransactionID, StudentID, Amount, TransactionDate, Type)
* **Sample Data:**
  + Students table with 100,000 records.
  + Finances table with 50,000 records.

**Azure Data Factory (ADF):**

1. **Linked Service:**
   * I configured a linked service to connect to the on-premise SQL Server.
   * **Details:**
     + I connected to the on-premise SQL Server using the linked service.
     + I validated the connection.
2. **Datasets:**
   * **I** created datasets for Students and Finances tables in ADF.
   * **Details:**
     + Define schema for Students and Finances datasets.
     + Ensure data types match between source and destination.
3. **Pipeline:**
   * **I did:** Create a pipeline with a Copy Activity to extract data from on-premise SQL Server and load it into Azure Data Lake Storage.
   * **Details:**
     + Configure source and destination in the Copy Activity.
     + Schedule the pipeline to run daily.
     + Monitor pipeline runs for errors and performance.
     + Example:
       - **Pipeline Name:** StudentFinanceMigrationPipeline
       - **Source:** On-premise SQL Server
       - **Destination:** Azure Data Lake Storage
       - **Schedule:** Daily at 2 AM

**Step 3: Data Ingestion and Transformation**

**Microsoft Fabric:**

* **I did:** Use Microsoft Fabric to orchestrate data pipelines and transformations.
  + **Details:**
    - Create a Data Factory within Microsoft Fabric.
    - Define data flows to transform and cleanse data.
    - Integrate with Azure Data Lake Storage and Synapse Analytics.
    - Example:
      * **Fabric Data Flow Name:** StudentFinanceDataFlow
      * **Transformations:** Remove null values, standardize date formats.

**Azure Data Lake Storage:**

* **I did:** Store raw data from the on-premise database.
  + **Details:**
    - Create containers for Students and Finances data.
    - Ensure data is stored in a structured format (e.g., CSV, Parquet).
    - Example:
      * **Container Names:** students-raw, finances-raw

**Azure Synapse Analytics:**

1. **SQL Pools:**
   * **I did:** Create dedicated SQL pools for data warehousing.
   * **Details:**
     + Create SynapseSQL.Students and SynapseSQL.Finances SQL pools.
     + Define table schemas matching source data.
     + Example:
       - **SQL Pool Names:** StudentsPool, FinancesPool
       - **Table Definitions:**

CREATE TABLE Students (StudentID INT, FirstName NVARCHAR (50), LastName NVARCHAR (50), DOB DATE, Email NVARCHAR (100), GPA FLOAT);

CREATE TABLE Finances (TransactionID INT, StudentID INT, Amount DECIMAL (10, 2), TransactionDate DATE, Type NVARCHAR (50));

1. **Stored Procedure:**

* **I did:** Create a stored procedure to calculate the total amount of financial aid received by each student.
* **Details:**
  + Example:

CREATE PROCEDURE CalculateTotalFinancialAid AS BEGIN SELECT StudentID, SUM(Amount) AS TotalAid FROM Finances WHERE Type = 'Scholarship' GROUP BY StudentID END;

* **Usage:** Execute this stored procedure to get total financial aid information.

EXEC CalculateTotalFinancialAid;

1. **Pipelines:**
   * **I did:** Transform data using Synapse pipelines.
   * **Details:**
     + Create data flow activities to transform data.
     + Cleanse data (e.g., remove null values, standardize formats).
     + Load transformed data into SQL pools.
     + Example:
       - **Pipeline Name:** TransformAndLoadPipeline
       - **Data Flow Activities:**
         1. Remove null values from Email column.
         2. Standardize date formats to YYYY-MM-DD.
         3. Load data into StudentsPool and FinancesPool.

**Azure Data Bricks:**

1. **Notebook:**
   * **I did:** Use PySpark in Data Bricks for data processing.
   * **Details:**
     + Create a notebook for data cleansing.
     + Example: Remove null values from the Email column in Students table.

from pyspark.sql import SparkSession spark = SparkSession.builder.appName("DataCleansing").getOrCreate() students\_df = spark.read.csv("path/to/students.csv", header=True) students\_df = students\_df.dropna(subset=["Email"]) students\_df.write.csv("path/to/cleaned\_students.csv", header=True)

**Step 4: Data Governance and Cataloging**

**Azure Purview:**

1. **Register Data Sources:**
   * **I did:** Register data sources in Azure Purview.
   * **Details:**
     + Register AzureSQLDB.Students and AzureSQLDB.Finances.
     + Example:
       - **Data Source Names:** StudentsSource, FinancesSource
2. **Scan Data Assets:**
   * **I did:** Scan data assets to ensure data lineage and compliance.
   * **Details:**
     + Configure scanning rules.
     + Schedule regular scans.
     + Example:
       - **Scan Frequency:** Weekly
       - **Scan Types:** Metadata scan, Data scan

**Step 5: Data Visualization**

**Power BI:**

1. **Dataset:**
   * **I did:** Connect Power BI to SynapseSQL.Students and SynapseSQL.Finances.
   * **Details:**
     + Import data from Synapse SQL pools into Power BI.
     + Example:
       - **Dataset Names:** StudentsDataset, FinancesDataset
2. **Report:**
   * **I did:** Develop a report on student performance metrics.
   * **Details:**
     + Create visuals for GPA distribution and financial aid disbursement.
     + Use DAX functions and M Queries for custom calculations.
     + Example: Calculate average GPA.

AverageGPA = AVERAGE(Students[GPA])

**Visuals:**

* **GPA Distribution:**
  + Histogram showing GPA ranges.
* **Financial Aid:**
  + Pie chart of financial aid types.
* **GPA Over Time:**
  + Line chart showing GPA trends over semesters.

**Sample Data and Reports:** Names are correct but rest information I changed due to privacy and only show here two records as sample.

**Students Table:**

| **StudentID** | **FirstName** | **LastName** | **DOB** | **Email** | **GPA** |
| --- | --- | --- | --- | --- | --- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | John | Doe | 1995-05-12 | john.doe@example.com | 3.5 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | Jane | Smith |  |  | 1996-07-23 | jane.smith@example.com | 3.8 |

**Finances Table:** Data is correct, but I only show two records.

| **TransactionID** | **StudentID** | **Amount** | **TransactionDate** | **Type** |
| --- | --- | --- | --- | --- |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1001 | 1 | 1500 | 10/22/2022 | Tuition |

|  |
| --- |
| 1002 2 500 10/22/2022 Scholarship |
| **Power BI Report:**   * **Dashboard Visuals:**   + **GPA Distribution:** Histogram showing GPA ranges.   + **Financial Aid:** Pie chart of financial aid types.   + **GPA Over Time:** Line chart showing GPA trends over semesters. |