Project Title: End-to-End Extended Warranty Management with Azure Data Factory, Azure Data Lake Gen 2, and Azure Databricks

Objective:

Create a solution that ingests extended warranty data, processes warranty claims and product lifecycle information, and stores processed results for analysis and reporting.

Tables Used:

1. **Raw Data Tables**:
    - **warranty_registrations**: Contains raw warranty registration data (product_id, warranty start/end dates).
    - **warranty_claims**: Contains raw warranty claims data (claim_id, product_id, claim_amount, claim_date).
    - **product_lifecycle**: Contains product lifecycle data (product_id, release_date, end_of_life_date).
2. **Processed Data Tables**:
    - **processed_warranty_data**: Contains transformed data with calculated **remaining_warranty_days**.
    - **warranty_claim_aggregates**: Contains aggregated claims data (product_id, region, total_claim_amount, claim_count).

Steps:

1. Data Ingestion with Azure Data Factory (ADF):

- **Create Linked Services**:
    - Linked Service for **Azure Data Lake Gen 2** to store raw data that will provide warranty registration, claims, and product lifecycle data.
- **Create a Pipeline**:
    - Use **Copy Activity** to extract data from various data sources (e.g., warranty registrations, claims, and product lifecycle data) into **Azure Data Lake Gen 2**.

2. Data Transformation and Validation with Azure Databricks:

- **Create a Databricks Workspace**:
    - Set up **Azure Databricks** workspace and cluster to process data. And created a Databricks notebook for data validation and transformation.
- **Data Validation**:
    - Ensure the data is clean and consistent by applying various validation rules before proceeding with transformations.

Validations Implemented:

1. **Date Validations**:
    - Ensure **warranty_start** is before **warranty_end**.
    - Ensure **claim_date** falls within the warranty period.
2. **Claim Amount Validations**:
    - Ensure **claim_amount** is greater than 0.
3. **Fraud Detection**:
    - Detect **multiple claims** for the same product by the same customer within a short period.
4. **Expired Warranties**:
    - Identify **expired warranties** where the **warranty_end** date is earlier than the current date.
5. **Data Integrity**:
    - Ensure that critical fields like **product_id**, **warranty_start**, and **claim_amount** are not null.

3. Orchestrating with Azure Data Factory:

- **Create ADF Pipeline** to automate the entire process:
    - **Step 1**: Use **Copy Activity** to ingest data into **Data Lake Gen 2**.
    - **Step 2**: Use **Databricks Notebook Activity** to trigger the Databricks notebook for processing and validating the data.

4. Storing Processed Data:

- After transformation, store the validated and processed data back into **Azure Data Lake Gen 2**.

Expected Outcome:

- **End-to-End Data Pipeline**: A fully automated pipeline that ingests warranty and claims data, processes it, and outputs actionable insights for better warranty management that help business users analyze warranty claims, identify fraud, track the remaining warranty periods of products, and optimize warranty services.

Possible Enhancements:

1. **Fraud Detection**: Enhance the transformation logic in Databricks to flag potentially fraudulent claims based on patterns such as unusually high claim amounts or multiple claims for the same product.
2. **Real-Time Processing**: Implement **Azure Stream Analytics** to process warranty data in near real-time (e.g., claims submission) and provide instant insights.
3. **Predictive Analytics**: Use **Azure Machine Learning** to predict the likelihood of warranty claims based on historical data and other features, allowing the business to proactively manage resources