SURAJIT SHOME
AZURE DATA ENGINEER
Phone: +31 647988248

2. Implement a data pipeline that:
2.1. Ingests a stocks.json file into a normalized structure. This json could include real time data, such as market information containing stock prices for some assets. (Real time?)
2.2. Ingests 2 CSVs (presumably the output of database):
2.2.1. Clients.csv: includes data from clients
2.2.2. Collaterals.csv: includes collateral credit information for some of the clients in Clients.csv. Some of these collaterals are stock assets that you can combine with the stocks.json file.
2.3. Create a target table called collateral_status which depicts an overview of the clients and their collaterals fluctuation based on the market value of their assets (stocks.json)
2.4. Save the resulting table

Ingesting the stocks.json File:

- You can use Azure Data Factory (ADF) to ingest the *stocks.json* file into Azure Databricks. Create a Databricks Notebook that reads the JSON data and processes it. You can pass parameters from ADF to the notebook using *dbutils.notebook.exit(myReturnValue).*
- In the notebook, parse the JSON data and normalize it into a structured format. We can use Python or Scala in your Databricks notebook to achieve this.

Ingesting Clients.csv and Collaterals.csv:

- Similar to the JSON file, create another ADF pipeline to ingest the Clients.csv and Collaterals.csv files into Azure Databricks.
- Process these CSV files in separate Databricks notebooks. You can use Spark DataFrame APIs to join the data and create a unified view.

Creating the Collateral Status Table:

- In your Databricks notebook, combine the data from the normalized JSON file and the joined CSV data.
- Calculate the market value of each collateral based on stock prices from the stocks.json file.
- Determine the fluctuation in collateral value over time.
- Create a target table called *collateral_status* that summarizes the clients and their collateral fluctuations.

SURAJIT SHOME
AZURE DATA ENGINEER
Phone: +31 647988248

Sample Snippet to do above:

```
# Assuming you have already loaded the data from stocks.json,
Clients.csv, and Collaterals.csv
# Calculate the market value of each collateral based on stock
prices
collaterals_df = ...  # Load Collaterals.csv into a DataFrame
stocks_df = ...  # Load stocks.json into a DataFrame

# Join the data to get relevant information
combined_df = collaterals_df.join(stocks_df,
on="stock_symbol", how="inner")

# Calculate the market value of each collateral
combined_df = combined_df.withColumn("market_value",
combined_df["quantity"] * combined_df["stock_price"])

# Determine the fluctuation in collateral value (you can
adjust this logic as needed)
combined_df = combined_df.withColumn("fluctuation",
combined_df["market_value"] - combined_df["initial_value"])

# Create the collateral_status table
collateral_status_df = combined_df.select("client_id",
"collateral_id", "fluctuation")

# Save the resulting table (you can choose an appropriate
storage location)
collateral_status_df.write.mode("overwrite").parquet("path/to/
your/storage/collateral_status")
print("Collateral status table created and saved
successfully!")
```

Saving the Resulting Table: Once you've computed the collateral status, save the resulting table to a suitable storage location. You can use Azure SQL Database, Azure Data Lake Storage, or any other storage service supported by ADF.