Use case: Data Ingestion and Transformation

**Scenario:** A company want to build a data pipeline to ingest and transform loan and loan repayment history data. Source data files are in JSON format. Transformed data will be used for reporting purpose.

Solution using Azure Data Factory:

**Data Sources:** Source files contain data about loans taken by customers, loan repayment history and customer details.

Source 1: JSON file for loan details stored in Azure Blob Storage(Loan.json)

Source 2: JSON file for loan repayment details stored in Azure Blob Storage(LoanRepayment.json)

Source 3: JSON file for customer details stored in Azure Blob Storage(Customer.json)

Data formats:

1. Customer.json

Fields: CustomerID, CustomerName, CustomerLocation {City, State, Country}

1. LoanRepayment.json file

Fields: LoanID, PaymentDate, PaymentAmount, PrincipalPayment, InterestPayment, PaymentReferenceNumber

1. Loan.json

Fields: LoanID, BorrowerID, LoanAccountNumber, LoanAmount, InterestRate, LoanTerm, LoanType, LoanStatus, OriginationDate, MonthlyPaymentDay, EMIAmount

Data files:



**Data Ingestion:**

Ingest data from above three JSON data files to databricks.

**Data Transformation:**

1. Calculate count of late EMI payment days and partial payment days for each loan id. Create two different columns: LateEMIPaymntDaysCnt, PartitialPymntDaysCnt
2. Read customer file data and divide field CustomerLocation in individual fields named ‘CustomerCity’, ’CustomerState’, ‘CustomerCountry’
3. Calculate principal balance payment left for each loan id. Monthly EMI amount consists of two parts: principal and interest. Principal balance payment left = total loan amount – total principal paid by customer.
4. Calculate total penalty paid by each customer. Penalty is levied when customer pays EMI after monthly EMI payment date. Penalty is 2.5% of monthly EMI amount.
5. Capitalize first letter of each word in field ‘CustomerName’
6. Add logic to remove row level duplicates from transformed dataframe.
7. If PaymentReferenceNumber is blank then replace it with NULL.

**Data Storage:**

Store transformed dataframe in Azure Data Storage in delta file format. Same file path will be used for writing data for all future pipeline runs. Please make sure to write data to target delta path using upsert(merge) logic so that if same data is given as input to pipeline in next run it won’t load any duplicates in given delta path. Use LoanID, BorrowerID as a upsert keys. If LoanID and BorrowerID in source record matches with target record then only below column values should be updated in target record else whole record should be inserted in delta file path,

Columns: LateEMIPaymntDaysCnt, PartitialPymntDaysCnt, TotalPenaltyPaid, PrincipalBalPymntLeft, CustomerCity, CustomerState, CustomerCountry

Output file structure:

Fields: LoanID, BorrowerID, CustomerName, LoanAccountNumber, LoanAmount, InterestRate, LoanTerm, LoanType, LoanStatus, OriginationDate, EMIAmount, LateEMIPaymntDaysCnt, PartitialPymntDaysCnt , TotalPenaltyPaid, PrincipalBalPymntLeft, CustomerCity, CustomerState, CustomerCountry

**Data Orchestration:**

Schedule the pipeline to run daily at a specific time to ingest and transform the latest data.

**Scheduling and Monitoring:**

Schedule the pipeline to run every day at 12 AM CST.

Monitor the pipeline execution through Azure Data Factory's monitoring interface, which provides information on the status, execution time, and any errors or warnings.

**Error Handling and Retry:**

Implement retry policies for failed activities to handle temporary issues such as network errors or resource unavailability.

**Data Quality and Validation:**

Perform data validation checks to ensure the integrity and quality of the transformed data, such as checking for missing or invalid values, data type consistency.

**Notification and Alerts:**

Configure email notifications to be sent to the data engineering team in case of pipeline failures or data quality issues.