Approach Document for Cloud-Based ETL Pipeline for Financial Analysis

Project Overview

In this project, we aim to create a cloud-based ETL (Extract, Transform, Load) pipeline to efficiently manage financial transaction data for analysis. This pipeline leverages **Talend Open Studio** for data integration, **Azure SQL Database** for data storage, and **Snowflake** as the data warehouse.

The project provides a scalable and reliable solution to extract data, filter for relevant information, and load it into a data warehouse for business insights.

Project Goals

1. **Accurate Data Extraction:** Ensure financial transaction data is reliably fetched from Azure.
2. **Data Transformation:** Clean, filter, and validate transaction records to prepare them for analysis.
3. **Data Loading:** Efficiently load transformed data into Snowflake.
4. **Automation and Monitoring:** Schedule and monitor ETL processes for consistent performance.

Solution Details

1. Data Extraction

We start by pulling the transaction data from an Azure SQL database.

Data Source Details:

- **Database:** Azure SQL Database containing financial transaction data.
- **Authentication:** Secure connection using Azure credentials.
- **Data Fields Extracted:**
    - Transaction ID

    - Customer ID
    - Transaction Date
    - Transaction Type
    - Amount
    - Transaction Status
    - Transaction Location
    - Currency

Talend Component:

- `tAzureSqlInput`: Connects to Azure SQL Database to retrieve transaction data.

SQL Query:

SELECT Transaction_ID, Customer_ID, Transaction_Date, Transaction_Type, Amount, Transaction_Status, Transaction_Location, Currency

FROM P4_TRANSACTION;

This query ensures we extract all relevant transaction data for processing.

---

2. Data Transformation

Once the data is extracted, it needs to be cleaned and filtered to retain only the relevant transactions.

Key Transformation Steps:

- **Filtering:** Only transactions with a status of "completed" are kept for analysis.
- **Data Type Conversion:** Dates and numeric values are standardized for consistency.
- **Validation:** Ensures all critical fields are complete and valid.

Talend Components Used:

- `tMap`: Maps input data to output schema and applies transformation logic.
- `tFilterRow`: Filters records to keep only "completed" transactions.
- `tConvertType`: Ensures data types match Snowflake's requirements.

Example Filter Logic:

Transaction_Status.equals("completed")

This logic ensures that only completed transactions move forward in the ETL pipeline.

---

3. Data Loading

The cleaned and validated data is loaded into a Snowflake table.

Snowflake table schema:

CREATE OR REPLACE TABLE transaction_temp (

Transaction_ID STRING,

Customer_ID STRING,

Transaction_Date DATE,

Amount FLOAT,

Currency STRING

);

Talend Components Used:

- `tSnowflakeConnection`: Establishes connection to Snowflake.
- `tSnowflakeOutput`: Inserts data into the target table.

Configuration Details:

- **Load Type:** Append only
- **Error Handling:** Logs any issues during data load for troubleshooting.

Pipeline Workflow

Data Flow Pipeline

[tAzureSqlInput] --> [tMap] --> [tFilterRow] --> [tSnowflakeOutput]

1. **Extract:** Pull transaction data using `tAzureSqlInput`.
2. **Transform:** Use `tMap` and `tFilterRow` to clean and filter the data.
3. **Load:** Insert data into Snowflake using `tSnowflakeOutput`.

Automation and Monitoring

To ensure the pipeline runs smoothly and automatically:

1. **Scheduling:**
    - Use Talend's scheduling features to automate the ETL process.
2. **Monitoring:**
    - Enable Talend logs and use Azure Monitor to track job performance and detect issues.

Testing and Validation

1. **Data Accuracy:** Ensure that only completed transactions are loaded.
2. **Performance Monitoring:** Test for optimal execution times.
3. **Error Management:** Simulate errors to validate error handling mechanisms.

Deliverables

1. Fully functional ETL pipeline using Talend, Azure, and Snowflake.
2. Filtered and validated transaction data available in Snowflake for analysis.
3. Documentation on pipeline design, workflow, and automation processes.

Potential Risks and Solutions

Risk	Mitigation Strategy
Connection Failures	Implement retry logic and monitor logs.
Data Inconsistencies	Use robust data validation techniques.

Performance Bottlenecks	Optimize SQL queries and Talend jobs.

Summary

By developing this cloud-based ETL pipeline, we enhance the efficiency of data integration and improve the accuracy of financial analysis. The result is a powerful tool that helps businesses gain insights from reliable, clean data.