Documentation: Data Transformation and Loading Using Python and Apache NiFi

Introduction

This document provides a detailed explanation of the solution implemented for the data transformation and loading task. The approach used is **Option 2: Python Scripts** from the provided assignment instructions. The solution involves the following steps:

- **Transforming raw data** from MySQL staging tables using Python and Pandas.
- **Creating dimension and fact tables** to support a star schema for analytical processing.
- **Orchestrating the workflow** using NiFi's ExecuteStreamCommand processor to run the Python scripts.

Approach Overview

The transformation and loading process is implemented in Python scripts, with NiFi used as an external execution tool. The workflow consists of:

1. **Data extraction:** Loading raw data from MySQL staging tables into Pandas DataFrames.
2. **Data transformation:** Applying necessary transformations using Pandas.
3. **Data loading:** Writing the transformed data back to MySQL in structured dimension and fact tables.
4. **Automation with NiFi:** Calling the Python script via NiFi's ExecuteStreamCommand processor.

Implementation Details

Transforming Data (transforming_tables.py)

This script is responsible for extracting, transforming, and loading data into MySQL tables. The key steps are:

1. **Connecting to MySQL:** A connection is established using SQLAlchemy:
2. **Extracting Data:** Raw tables are loaded into Pandas DataFrames:
3. **Applying Transformations:** Each table undergoes specific transformations as required:
    - **Payments Table**
        - Aggregates payment information per order.
        - Capitalizes PaymentType and replaces underscores with spaces.
    - **Feedback Table**
        - Converts FeedbackScore to integer.

    - Formats FeedbackFormSentDate and FeedbackAnswerDate as datetime.
- **Users Table**
    - Capitalizes UserCity and UserState values.
- **Orders Table**
    - Joins feedback data with orders using OrderID.

4. **Loading Transformed Data:** The transformed tables are saved back to MySQL

Creating Dimension and Fact Tables (star_schema.py)

This script builds structured tables for analytics.

Creating Dimension Tables

Each dimension table is created from the transformed data. Dimension tables include:

- dim_users
- dim_feedbacks
- dim_payments
- dim_products
- dim_sellers
- dim_date
- dim_time

2. Creating the Fact Table

The fact_order_items table centralizes transactional data. Data is then inserted into fact_order_items using SQL joins on transformed tables.

Automating Execution with NiFi

NiFi is used to call the Python scripts automatically.

Setting Up ExecuteStreamCommand

- The **ExecuteStreamCommand** processor is configured to run
- This ensures NiFi triggers the Python script as part of the data pipeline.

Conclusion

This implementation successfully extracts, transforms, and loads data into a structured format using Python, with NiFi handling execution automation. This approach ensures efficient transformation and scalable data processing for analytics.