Low Level Design (LLD)

Project: EGX30 Stock Data Pipeline

Technology Stack: Apache Airflow, Apache Hive

Date: 21-Dec-2025

Prepared By: Marwa Mansour

Code Details

DAG Name

- **egx30_stock_pipeline**

DAG Responsibility

- Orchestrates the end-to-end data pipeline for EGX30 stock prices.
- Executes tasks sequentially: Extract → Validate → Load.

Tasks Description

Task Name	Responsibility
extract_data	Fetch daily stock prices from Yahoo Finance
validate_and_prepare	Validate, clean, deduplicate, and prepare data
load_to_hive	Load processed data into Hive

Task Dependency

- Linear dependency ensures data consistency:

Extract → Validate → Load

DDLs (Hive Tables Design)

2.1 Staging Table

Purpose:

- Temporary storage for raw validated data before transformation.

Design Characteristics:

- Format: TEXTFILE
- Not partitioned
- Used for fast loading and reprocessing

Columns:

- stock_symbol (STRING)
- price (DECIMAL)

Why Staging Table?

- Isolates raw data
- Simplifies debugging
- Allows safe reprocessing

2.2 Final Table

Purpose:

- Optimized table for analytics and reporting.

Design Characteristics:

- Format: ORC
- Partitioned by trade date
- Compressed for performance

Columns:

- stock_symbol (STRING)
- price (DECIMAL)

Partition Column:

- trade_date (STRING)

Benefits:

- Faster query execution
- Reduced storage
- Efficient historical analysis

Connection Details

Airflow Connections

Component	Connection Type
Airflow Scheduler	Local Executor
Python Tasks	Local Python Environment
Hive	Hive CLI

External Connections

- **Yahoo Finance API**
  - Protocol: HTTPS
  - Authentication: None (Public API)
  - Data Format: JSON

File System Usage

- Temporary storage path:

/tmp/
Used for:

- CSV files
- Hive-ready data files
- HQL scripts

Integration Details

Airflow ↔ Yahoo Finance

- Uses REST API calls to fetch stock data.
- Data returned in JSON format.
- Extracts daily closing price.

Airflow ↔ Hive

- Hive queries executed using Bash Operator.
- Dynamic HQL scripts generated per execution date.
- Partition-based loading ensures idempotency.

Inter-Task Communication

- **XCom**
  - Used to pass file paths between tasks.
  - Avoids data duplication.

Mechanism (Pipeline Workflow)

Step-by-Step Execution Flow

1. **Trigger**
   - DAG runs daily at scheduled time.
2. **Extraction**
   - Fetches daily stock prices.
   - Handles API failures gracefully.
   - Stores results in CSV format.
3. **Validation**
   - Verifies data types.
   - Removes invalid records.
   - Deduplicates stock symbols.
4. **Preparation**
   - Converts data into Hive-compatible format.
   - Generates dynamic Hive scripts.

5. **Loading**
    - Loads data into staging table.
    - Inserts data into partitioned final table.
    - Drops old partitions to support reruns.
6. **Verification**
    - Confirms record count per partition.

Reliability & Fault Tolerance

- Automatic retries on task failure.
- Logs available in Airflow UI.
- Failure in one task prevents downstream execution.
