Wikipedia Web Scrape Data Pipeline

Source	Raw	Staging	Warehousing	User

Raw
- Python script performs web scrape on Wikipedia website that contains relevant company names and information fro S&P 500
- Raw data is loaded into database hosted on AWS

Staging
- Reads from raw_wikipedia_sp500 table.
- Selects relevant columns.
- Performs basic cleaning (e.g., TRIM, UPPER on symbol).
- Creates the stg_wikipedia_scrape view.

Warehousing
- Reads from staging views (stg_wikipedia_scrape).
- Generates surrogate keys (e.g., date_id, symbol_id).
- Builds dimensional tables (dim_date, dim_symbol).
- Joins dimensions to create the fact_price table. Creates the final star schema structure.

Wikipedia_Web_Scrape_Extract_Load_Python.py

stg_wikipedia_web_scrape.sql

dim_date.sql
dim_symbol.sql
fact_price.sql

Website

Shema: sql_project
Table(s): raw_wikipedia_sp500

Shema: sql_project
Views: stg_wikipedia_web_scrape

Shema: sql_project
Table(s): dim_symbol
dim_date

GitHub Actions
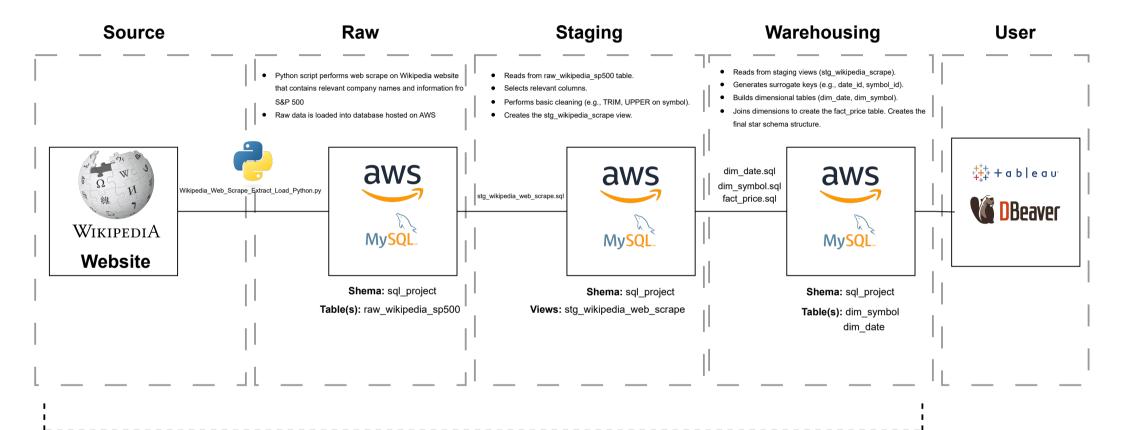
dbt_pipeline.yml

tiingo_api_pipeline.yml

wikipedia_web_scrape_pipeline.yml

- tiingo_api_pipeline.yml:
  - Runs on a schedule (or manually).
  - Executes Python script to fetch latest stock prices from Tiingo API.
  - Loads fetched price data into the raw_prices table in MySQL.
- wikipedia_web_scrape_pipeline.yml:
  - Runs on a schedule (or manually).
  - Executes Python script to scrape the S&P 500 company list from Wikipedia.
  - Loads scraped company data into the raw_wikipedia_sp500 table in MySQL.
- dbt_pipeline.yml:
  - Runs on a schedule (likely after data ingestion) or manually.
  - Executes dbt run command.
  - Transforms data from raw tables -> staging views -> warehouse tables within MySQL.