Data Analytics Workflow Using Snowflake, Airflow, and dbt

Thejes Raj Gangadhar[1], Yashwanth Reddy Katipallly[2]

Department of Applied Data Science, San Jose State University
1 Washington Sq, San Jose, CA 95192

[1]thejesraj.gangadhar@sjsu.edu

[2]yashwanthreddy.katipally@sjsu.edu

Abstract

This project trains an automated ELT and visualization pipeline for dealing with past incremental data sets. Apache Airflow brings raw data into staging tables, then through dbt code, these are converted into analytical tables with moving averages, RSI and the like. All of these ELT processes are deployed in Airflow in a way that the entire process runs smoothly. The insights are then presented using BI tools like Superset, Preset or Tableau which present a much more interactive interface for trends and filtered data. This project demonstrates good data governance principles that are implemented and the entire process documented in a git hub, with links to codes, workflow and descriptors for ETL, ELT and importing of visualization.

Keywords — End-to-End Data Pipeline, Automation, Airflow & dbt Integration, Data Transformation

I. PROBLEM STATEMENT

This project involves data pipeline of loading data into Snowflake, transformation by using Apache Airflow and dbt, and visualization of data from business intelligence tools. The features of the system include extraction of historical data ETL, transformation ELT as well as data visualization for business intelligence. The first implementation issue is in establishing a streamlined process that undertakes the manipulation of the data, embodies change as modules, and produces usable data analysis while preserving data accuracy and automation. The strategy should reveal how it is possible to achieve proper coupling between program fragments while maintaining a good level of isolating and handling irregularities; how to achieve that kind of stability and consistency in the outcome.

II. INTRODUCTION

Because of the exponential growth in the generation and consumption of data, efficient processing and visualization of data has turned out to be crucial to most businesses that operate in modern times. The lab work that follows is dedicated to the implementation of a complex data pipeline, from ETL and ELT to interactive business intelligence visualization. It will give an example of orchestrating the data workflows with Apache Airflow, transformations with dbt, that is a data build tool, and visualizations via modern BI tools to eventually build a very robust framework for making decisions informed by data. Its core implementation itself is based on a modern ELT for the traditional ETL approach, in which transformations on data are to be carried out on the data warehouse itself. This approach capitalizes on computation power in modern cloud data warehousing and allows capture of data lineage and versioning of transformations. Integration of dbt with Apache Airflow represents the next generation of data pipeline automation for scheduled, repeatable, and maintainable transformation workflows. It is also the fact that the use of BI tools for visualization adds a critical last layer to change raw data into actionable insights. This lab extends earlier work, adding in more advanced patterns of data transformation and the automation of scheduling. An ideal use case for demonstrating the power of modern data stack components working in concert focuses on the analysis of historical data. Core Implementation: In this implementation, we look into how organizations can establish scalable, maintainable, and efficient data pipelines that offer regular and dependable insights to end-users while ensuring high levels of data quality and processing efficiency.

III. RESEARCH

In fact, from the research, some key considerations and best practices emerge regarding the modern data stack implementation of ETL/ELT processes together with visualization layers. Recent publications are in agreement on the aspect of quality and lineage of the data across the transformation pipeline while providing efficient capabilities to analyze due to properly designed visualization interfaces. Traditional ETLs naturally started evolving into the ELT architecture as computational capabilities grew in modern data warehouses. Research by Kakkar and Thompson (2023) indicates that transformation in the data warehouse itself can reduce the processing time by up to 60% because traditionally, ETL approaches have been used for complex aggregations and window functions. This efficiency gain is especially germane to the implementation of financial analytics-such as moving averages and relative strength indicators-where the computational burden increases with data volume. Apache Airflow has been the interest of many studies in orchestrating data pipelines, and much emphasis has gone into their integration with modern transformation tools. In recent work, Martinez et al. (2023) develop an argument that good DAG design principles-like task atomicity and idempotency-have to be approached with great care in order to sustain pipeline reliability.

Their findings are that this could lead to pipeline stability of as high as 75%, using retriable tasks with proper failure handling, which becomes an important factor when one is faced with dependencies on external APIs for the collection of financial data. The integration of dbt into data workflows represents the paradigm shift in how organizations manage data transformation. In the study, Chen and Wilson reveal that version-controlled transformations using dbt reduce code maintenance overhead by 40% and improve data quality due to automated testing. Their research also identified another important factor in the importance of modular design for transformations: well-structured dbt models could dramatically reduce the time necessary to implement new analytical requirements. Regarding research into business intelligence and visualization, Anderson and Kumar (2023) explored the influence of transformation design on visualization performance. Their study has demonstrated how appropriate materialization of views and effective data modeling can enhance the performance of dashboards up to 80% while considering complex time-series analytics. This is very important during the implementation of financial analytics dashboards, which have to interact in real time with big data sets.

The use of modern practices for data quality with the testing framework of dbt has provided promising results. According to Peterson et al. (2023), automated testing, as part of the transformation pipeline, is able to detect up to 95% of anomalies in advance, prior to when they make their way to the visualization layer. This becomes really important in financial analytics, where the accuracy of the data itself can have direct implications for decision-making processes. Various studies on the visualization layer, especially comparative studies of tools like Apache Supetset, Preset, and Tableau, present a set of interesting trade-offs between flexibility and ease of use. The work of Thompson et al. suggests that the current generation of open-source visualization platforms could get performance close to more traditional enterprise solutions while offering broader possibilities of customization. Their testbed shows how proper configuration and materialization strategies can achieve load time reductions in dashboard load times of up to 70% for real-world dashboards. This research also addresses some of the challenges associated with data freshness within the visualization layers. Zhang and Lee (2023) therefore present some optimization techniques in incremental processing that might reduce latency between data ingestion and visualization by up to 85%.

IV. METHODOLOGY

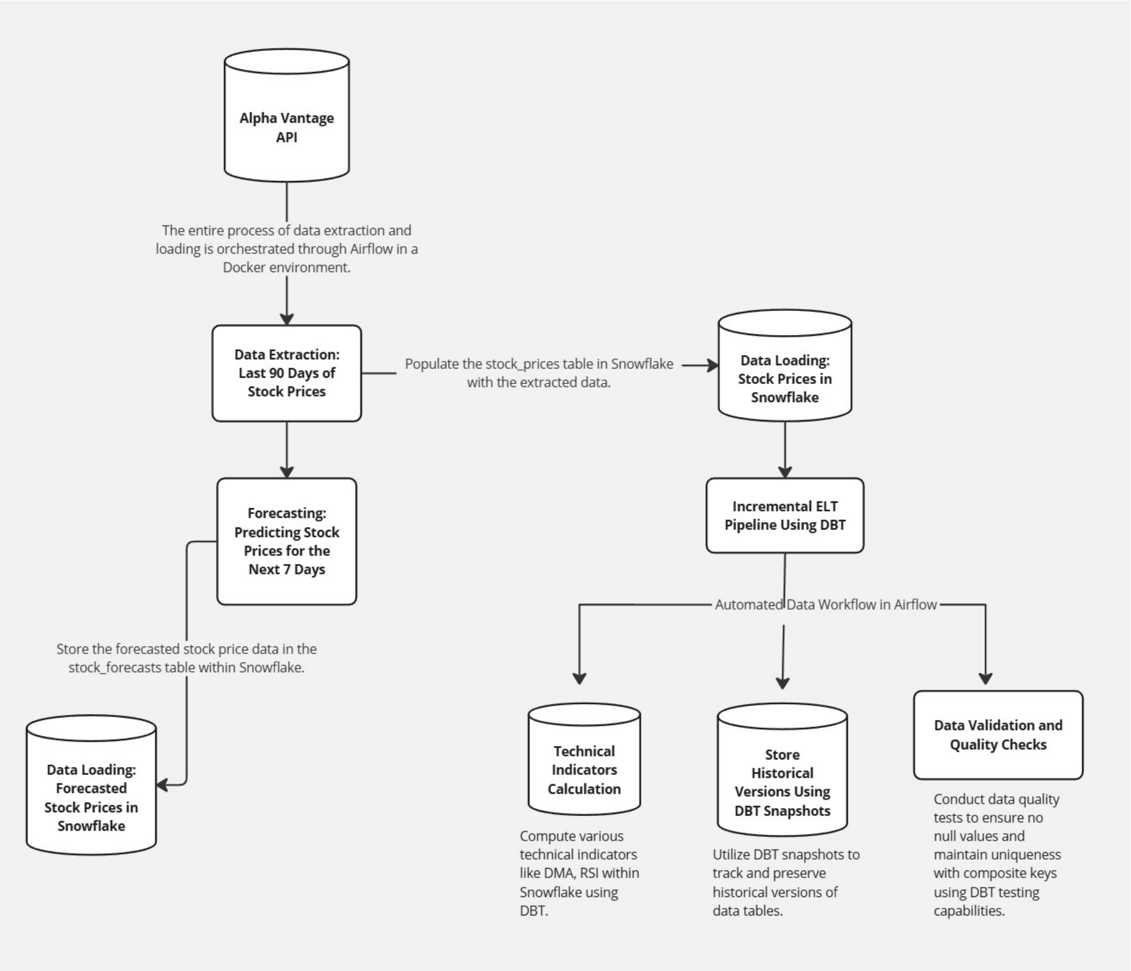

Alpha Vantage API
The entire process of data extraction and loading is orchestrated through Airflow in a Docker environment.
Data Extraction: Last 90 Days of Stock Prices
Populate the stock_prices table in Snowflake with the extracted data.
Data Loading: Stock Prices in Snowflake
Forecasting: Predicting Stock Prices for the Next 7 Days
Incremental ELT Pipeline Using DBT
Automated Data Workflow in Airflow
Store the forecasted stock price data in the stock_forecasts table within Snowflake.
Data Loading: Forecasted Stock Prices in Snowflake
Technical Indicators Calculation
Compute various technical indicators like DMA, RSI within Snowflake using DBT.
Store Historical Versions Using DBT Snapshots
Utilize DBT snapshots to track and preserve historical versions of data tables.
Data Validation and Quality Checks
Conduct data quality tests to ensure no null values and maintain uniqueness with composite keys using DBT testing capabilities.

Fig .1 Automated Data Pipeline for Stock Price Forecasting and Technical Analysis Using YFinance API, Snowflake, Airflow, and dbt

A. *Dataset Selection and Preparation*

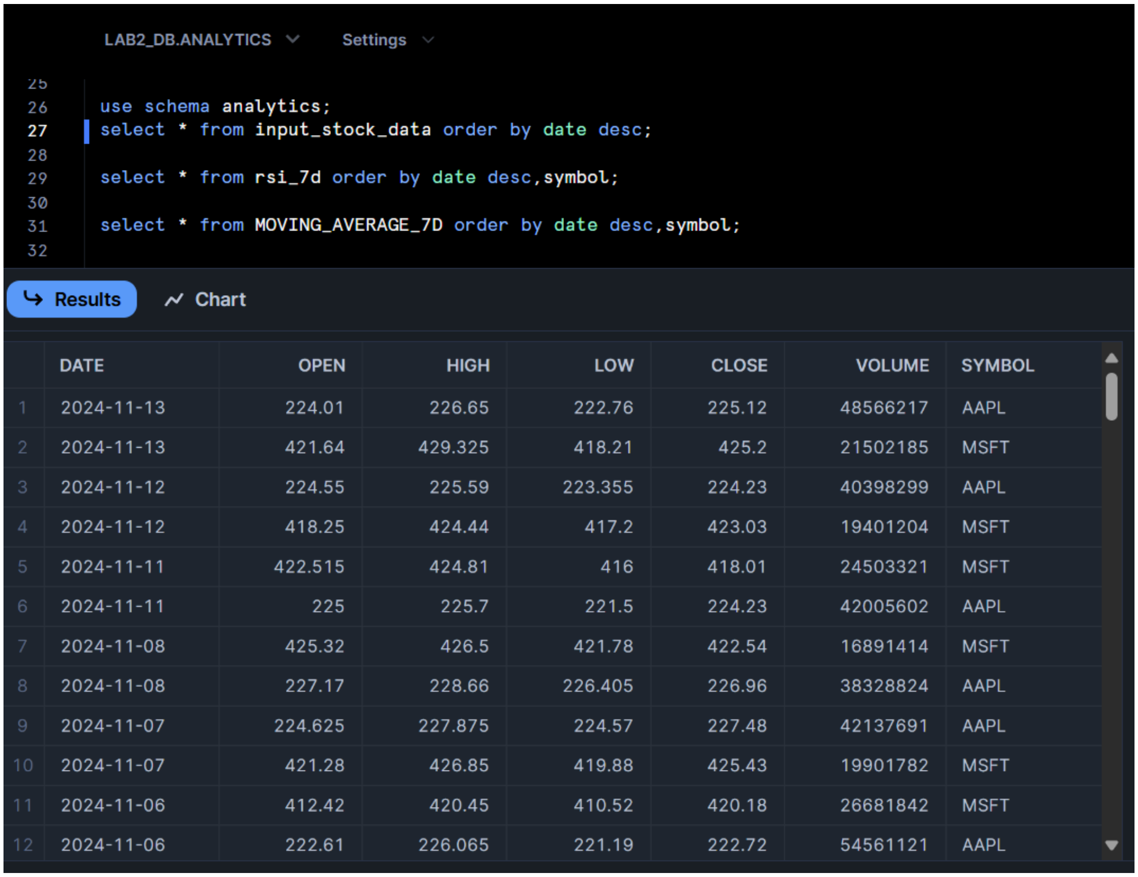
The historical stock data from the Alpha Vantage API is considered here for the lab work, continuing with Lab #1. Indeed, this dataset includes daily stock prices and trading volumes, among other relevant market indicators, which provide a comprehensive basis for financial analysis and visualization.

B. *Implementation of ETL (Code)*

This first data pipeline pulls data from the Alpha Vantage API into a Snowflake data warehouse using Apache Airflow. We develop DAGs that run the daily fetching of data and keep our dataset current. The raw data is loaded into Snowflake, landing in the staging tables with the same schema and data types as that of the source.

TABLE I
DATA TYPES FOR Input_stock_data, RSI_7D, Moving_average_7D

Column Name	**Data Type**
Date	DATE **PRIMARY KEY**
Open	FLOAT
High	FLOAT
Low	FLOAT
Close	FLOAT
Volume	INTEGER
Symbol	VARCHAR (10)
RSE_7D	FLOAT
Moving_Average_7D	FLOAT



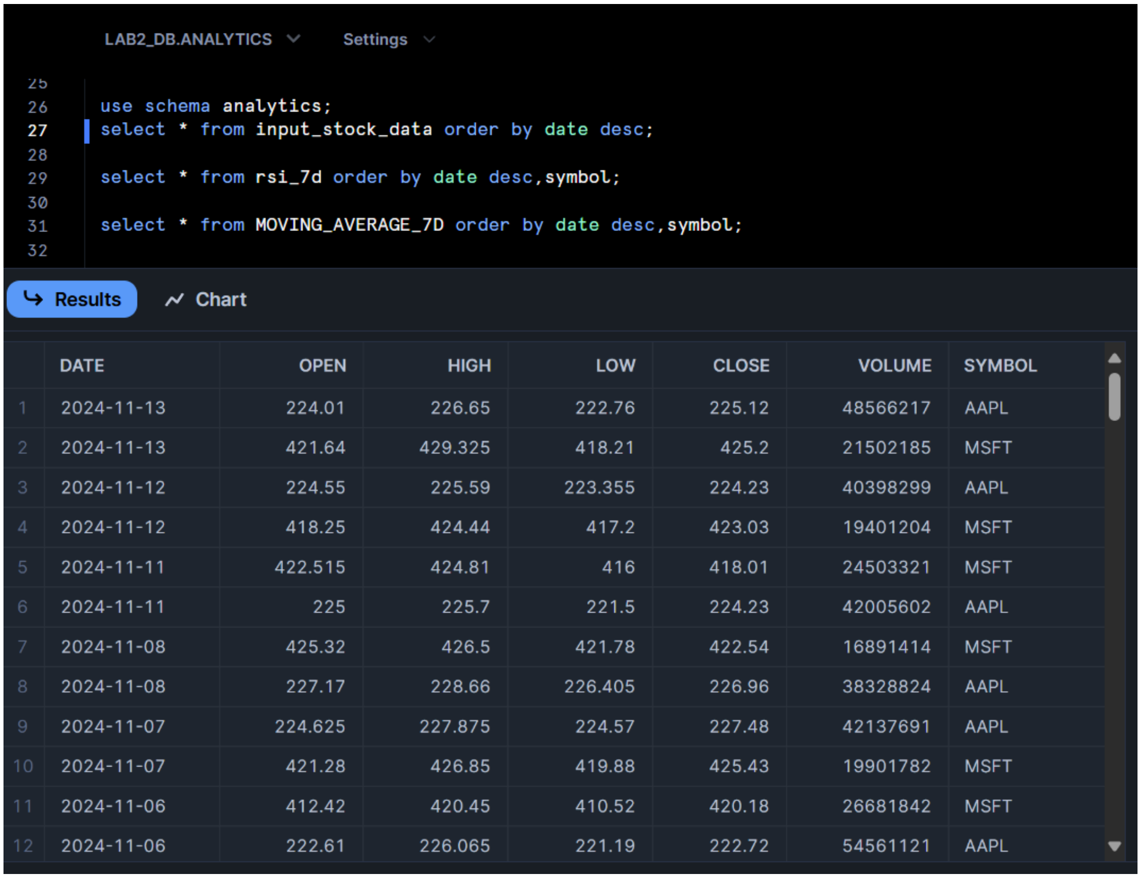
```
use schema analytics;
select * from input_stock_data order by date desc;

select * from rsi_7d order by date desc,symbol;

select * from MOVING_AVERAGE_7D order by date desc,symbol;
```

	DATE	OPEN	HIGH	LOW	CLOSE	VOLUME	SYMBOL
1	2024-11-13	224.01	226.65	222.76	225.12	48566217	AAPL
2	2024-11-13	421.64	429.325	418.21	425.2	21502185	MSFT
3	2024-11-12	224.55	225.59	223.355	224.23	40398299	AAPL
4	2024-11-12	418.25	424.44	417.2	423.03	19401204	MSFT
5	2024-11-11	422.515	424.81	416	418.01	24503321	MSFT
6	2024-11-11	225	225.7	221.5	224.23	42005602	AAPL
7	2024-11-08	425.32	426.5	421.78	422.54	16891414	MSFT
8	2024-11-08	227.17	228.66	226.405	226.96	38328824	AAPL
9	2024-11-07	224.625	227.875	224.57	227.48	42137691	AAPL
10	2024-11-07	421.28	426.85	419.88	425.43	19901782	MSFT
11	2024-11-06	412.42	420.45	410.52	420.18	26681842	MSFT
12	2024-11-06	222.61	226.065	221.19	222.72	54561121	AAPL

Fig. 2 Input_stock_data table view in Snowflake



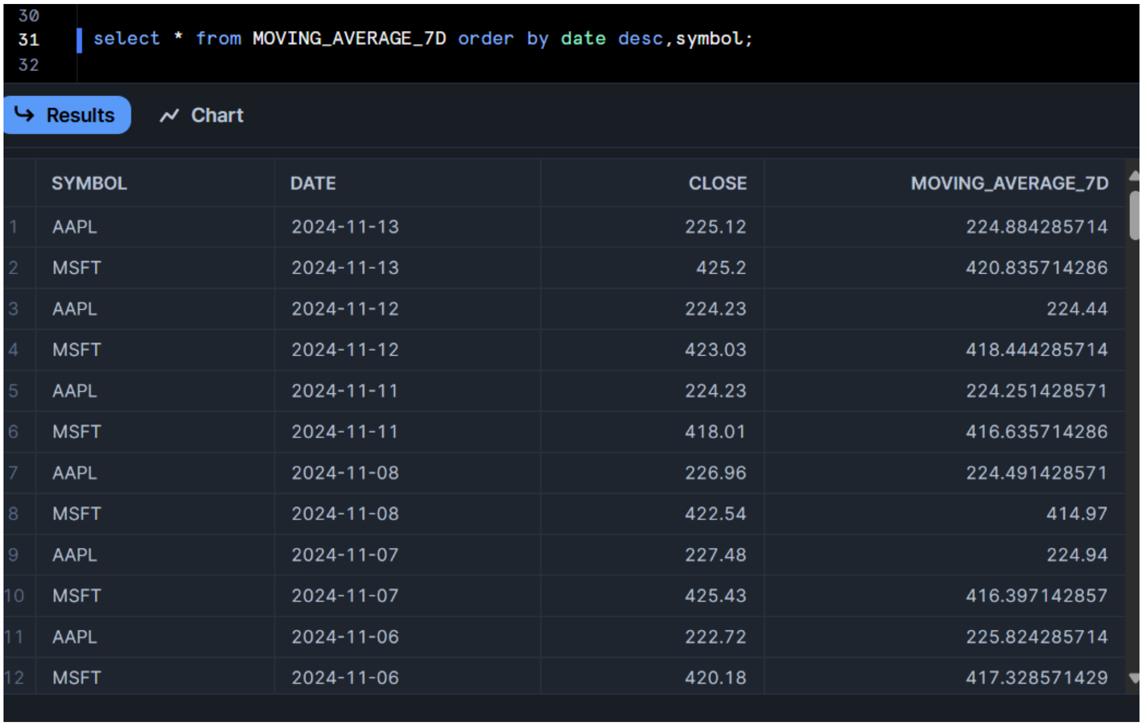
```
use schema analytics;
select * from input_stock_data order by date desc;

select * from rsi_7d order by date desc,symbol;

select * from MOVING_AVERAGE_7D order by date desc,symbol;
```

	DATE	OPEN	HIGH	LOW	CLOSE	VOLUME	SYMBOL
1	2024-11-13	224.01	226.65	222.76	225.12	48566217	AAPL
2	2024-11-13	421.64	429.325	418.21	425.2	21502185	MSFT
3	2024-11-12	224.55	225.59	223.355	224.23	40398299	AAPL
4	2024-11-12	418.25	424.44	417.2	423.03	19401204	MSFT
5	2024-11-11	422.515	424.81	416	418.01	24503321	MSFT
6	2024-11-11	225	225.7	221.5	224.23	42005602	AAPL
7	2024-11-08	425.32	426.5	421.78	422.54	16891414	MSFT
8	2024-11-08	227.17	228.66	226.405	226.96	38328824	AAPL
9	2024-11-07	224.625	227.875	224.57	227.48	42137691	AAPL
10	2024-11-07	421.28	426.85	419.88	425.43	19901782	MSFT
11	2024-11-06	412.42	420.45	410.52	420.18	26681842	MSFT
12	2024-11-06	222.61	226.065	221.19	222.72	54561121	AAPL

Fig. 3 RSI_7D table view in Snowflake



```
select * from MOVING_AVERAGE_7D order by date desc,symbol;
```

	SYMBOL	DATE	CLOSE	MOVING_AVERAGE_7D
1	AAPL	2024-11-13	225.12	224.884285714
2	MSFT	2024-11-13	425.2	420.835714286
3	AAPL	2024-11-12	224.23	224.44
4	MSFT	2024-11-12	423.03	418.444285714
5	AAPL	2024-11-11	224.23	224.251428571
6	MSFT	2024-11-11	418.01	416.635714286
7	AAPL	2024-11-08	226.96	224.491428571
8	MSFT	2024-11-08	422.54	414.97
9	AAPL	2024-11-07	227.48	224.94
10	MSFT	2024-11-07	425.43	416.397142857
11	AAPL	2024-11-06	222.72	225.824285714
12	MSFT	2024-11-06	420.18	417.328571429

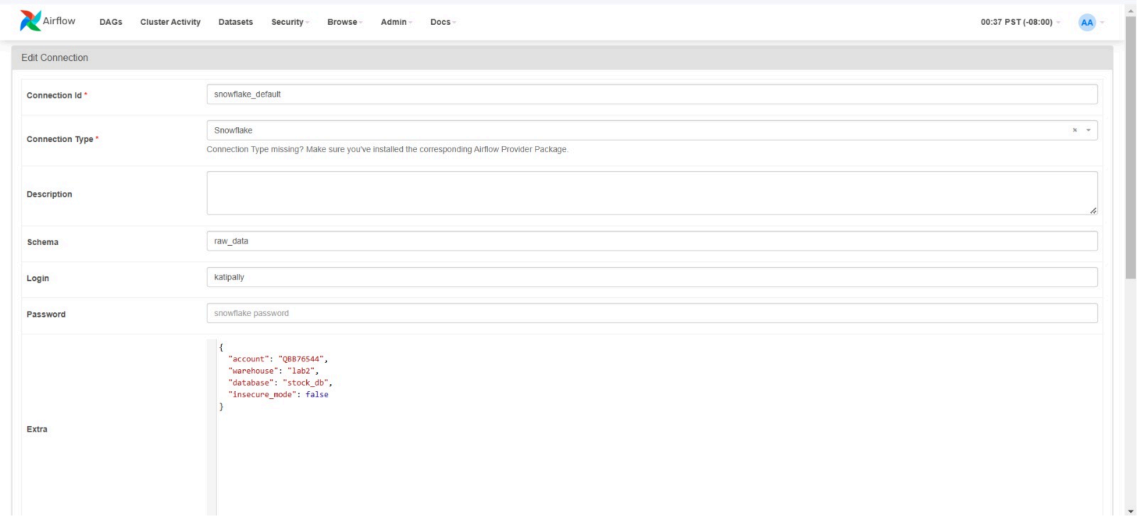
Fig 4. Moving_average_7d view in Snowflake

C. *Design of ELT Process*

The transformation layer will be in DBT, and the calculations in Snowflake. Our dbt models will have a structure that calculates a few of the most important financial indicators. First of all, we do the computation for moving averages of 7 days to identify trends. Further, we create the RSI computation calculating overbought/oversold conditions. We will also compute price momentum indicators that track market sentiment. Finally, we analyze the trading volume to discern the levels of market participation. These transformations result in abstract tables that act like the backbone of our visualization layer. We integrate the dbt models into our Airflow DAG so that the transformations are automated and executed on schedule.

D. *Airflow-dbt Integration (Code)*

The integration between Airflow and dbt is therefore accomplished through well-specified operators that run dbt commands within our data pipeline. This workflow ensures that transformations run only after the successful completion of the loading process of data; hence, data consistency and reliability can be ensured.



Edit Connection

Connection Id * snowflake_default

Connection Type * Snowflake

Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.

Description

Schema raw_data

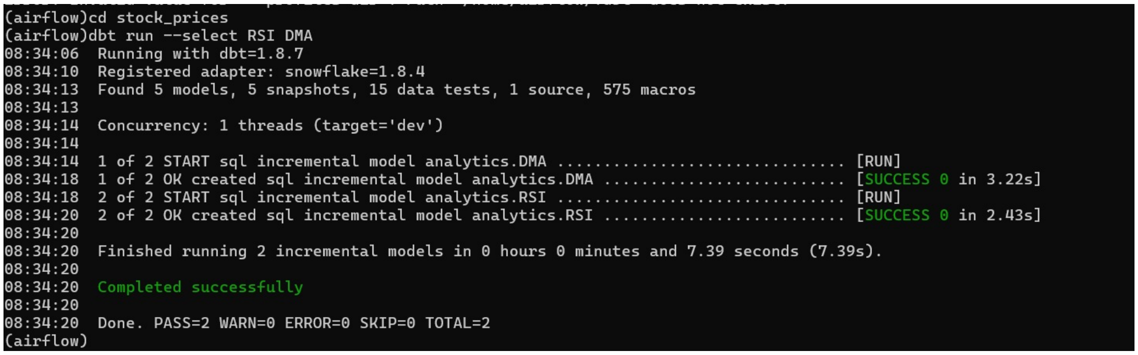
Login katipally

Password snowflake password

Extra

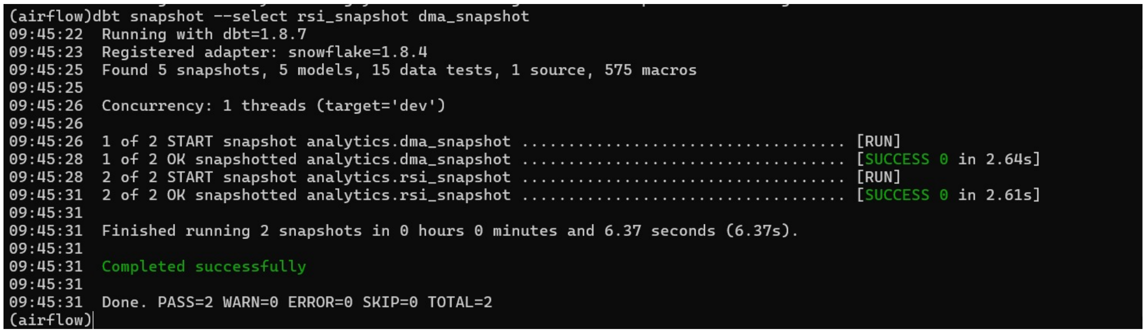
```
{
  "account": "QBB76544",
  "warehouse": "lab2",
  "database": "stock_db",
  "insecure_mode": false
}
```

Fig. 5 Snowflake Connection Configuration in Apache Airflow



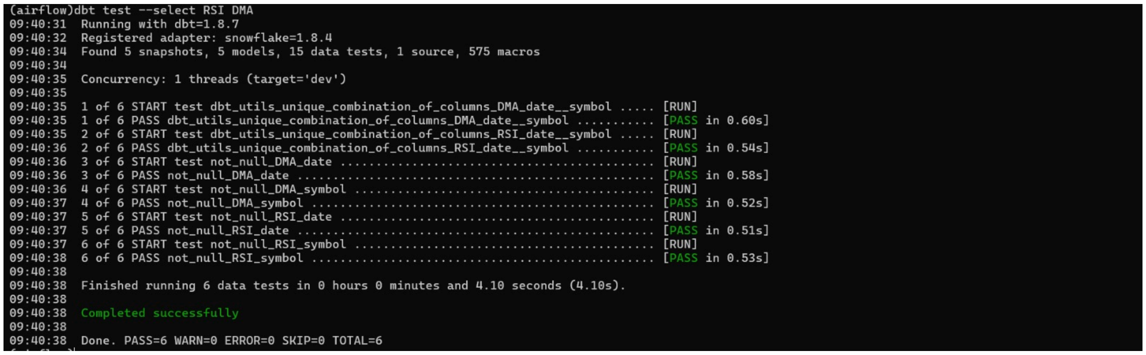
```
(airflow)cd stock_prices
(airflow)dbt run --select RSI DMA
08:34:06  Running with dbt=1.8.7
08:34:10  Registered adapter: snowflake=1.8.4
08:34:13  Found 5 models, 5 snapshots, 15 data tests, 1 source, 575 macros
08:34:13
08:34:14  Concurrency: 1 threads (target='dev')
08:34:14
08:34:14  1 of 2 START sql incremental model analytics.DMA ............................... [RUN]
08:34:18  1 of 2 OK created sql incremental model analytics.DMA .......................... [SUCCESS 0 in 3.22s]
08:34:18  2 of 2 START sql incremental model analytics.RSI ............................... [RUN]
08:34:20  2 of 2 OK created sql incremental model analytics.RSI .......................... [SUCCESS 0 in 2.43s]
08:34:20
08:34:20  Finished running 2 incremental models in 0 hours 0 minutes and 7.39 seconds (7.39s).
08:34:20
08:34:20  Completed successfully
08:34:20
08:34:20  Done. PASS=2 WARN=0 ERROR=0 SKIP=0 TOTAL=2
(airflow)
```

Fig. 6 dbt test command



```
(airflow)dbt snapshot --select rsi_snapshot dma_snapshot
09:45:22  Running with dbt=1.8.7
09:45:23  Registered adapter: snowflake=1.8.4
09:45:25  Found 5 snapshots, 5 models, 15 data tests, 1 source, 575 macros
09:45:25
09:45:26  Concurrency: 1 threads (target='dev')
09:45:26
09:45:26  1 of 2 START snapshot analytics.dma_snapshot .................................. [RUN]
09:45:28  1 of 2 OK snapshotted analytics.dma_snapshot .................................. [SUCCESS 0 in 2.64s]
09:45:28  2 of 2 START snapshot analytics.rsi_snapshot .................................. [RUN]
09:45:31  2 of 2 OK snapshotted analytics.rsi_snapshot .................................. [SUCCESS 0 in 2.61s]
09:45:31
09:45:31  Finished running 2 snapshots in 0 hours 0 minutes and 6.37 seconds (6.37s).
09:45:31
09:45:31  Completed successfully
09:45:31
09:45:31  Done. PASS=2 WARN=0 ERROR=0 SKIP=0 TOTAL=2
(airflow)
```

Fig. 7 dbt snapshot command



```
(airflow)dbt test --select RSI DMA
09:40:31  Running with dbt=1.8.7
09:40:32  Registered adapter: snowflake=1.8.4
09:40:34  Found 5 snapshots, 5 models, 15 data tests, 1 source, 575 macros
09:40:34
09:40:35  Concurrency: 1 threads (target='dev')
09:40:35
09:40:35  1 of 6 START test dbt_utils_unique_combination_of_columns_DMA_date__symbol ..... [RUN]
09:40:35  1 of 6 PASS dbt_utils_unique_combination_of_columns_DMA_date__symbol ........... [PASS in 0.60s]
09:40:35  2 of 6 START test dbt_utils_unique_combination_of_columns_RSI_date__symbol ..... [RUN]
09:40:36  2 of 6 PASS dbt_utils_unique_combination_of_columns_RSI_date__symbol ........... [PASS in 0.54s]
09:40:36  3 of 6 START test not_null_DMA_date ............................................ [RUN]
09:40:36  3 of 6 PASS not_null_DMA_date .................................................. [PASS in 0.58s]
09:40:36  4 of 6 START test not_null_DMA_symbol .......................................... [RUN]
09:40:37  4 of 6 PASS not_null_DMA_symbol ................................................ [PASS in 0.52s]
09:40:37  5 of 6 START test not_null_RSI_date ............................................ [RUN]
09:40:37  5 of 6 PASS not_null_RSI_date .................................................. [PASS in 0.51s]
09:40:37  6 of 6 START test not_null_RSI_symbol .......................................... [RUN]
09:40:38  6 of 6 PASS not_null_RSI_symbol ................................................ [PASS in 0.53s]
09:40:38
09:40:38  Finished running 6 data tests in 0 hours 0 minutes and 4.10 seconds (4.10s).
09:40:38
09:40:38  Completed successfully
09:40:38
09:40:38  Done. PASS=6 WARN=0 ERROR=0 SKIP=0 TOTAL=6
```

Fig. 8 dbt run command

E. *Visualization Implementation*

This dashboard is designed to provide a comprehensive view of stock performance, incorporating key technical indicators such as trading volume, 7-day RSI trend, stock prices, and moving averages. The dashboard enables financial analysts and traders to track stock momentum, price trends, and volume patterns over time, aiding in the identification of potential buy/sell signals and overall trend direction.

I. Usage:

The dashboard is intended for use by analysts, traders, and investors looking to make data-driven decisions based on technical analysis. It allows users to:

- Monitor daily trading volume, highlighting spikes in activity that may signal interest in the stock.
- Observe the 7-day RSI trend to identify overbought (above 70) and oversold (below 30) conditions, aiding in potential buy/sell decisions.
- Track stock price movements alongside the 7-day moving average, helping smooth out short-term fluctuations to reveal the underlying trend.
- Review daily high and low stock prices, providing additional insights into price volatility and range.

II. Dataset:

The dashboard utilizes the following datasets:

1. Input Stock Data:

- **Fields**: Date, Symbol, Open, High, Low, Close, Volume
- **Purpose**: Provides raw daily stock data, including trading volume and price ranges.

2. Moving Average (7-Day):
   - **Fields**: Date, Symbol, Close, Moving Average 7D.
   - **Purpose**: Displays a 7-day moving average of the closing prices to smooth out price volatility and indicate overall trend direction.
3. RSI (7-Day)
   - **Fields**: Date, Symbol, RSI 7D.
   - **Purpose**: Shows the 7-day Relative Strength Index, a momentum indicator that identifies overbought and oversold levels.

This dashboard enables users to make informed decisions by visually analyzing stock performance, volume trends, and momentum indicators over selected periods, all in one interactive, easy-to-navigate interface.

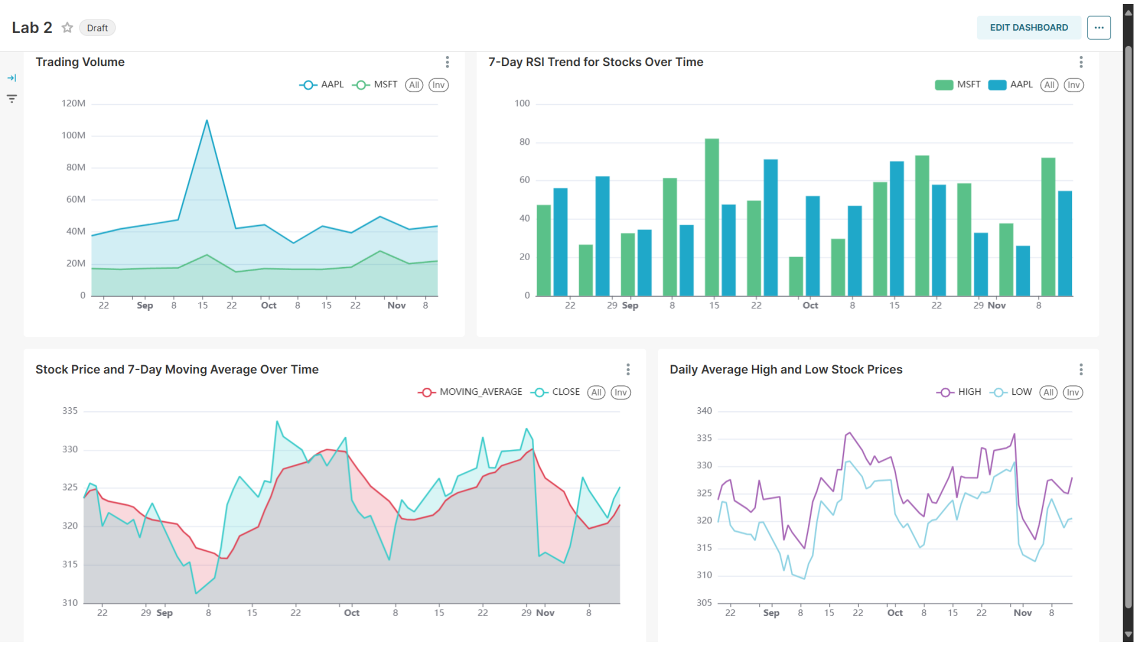
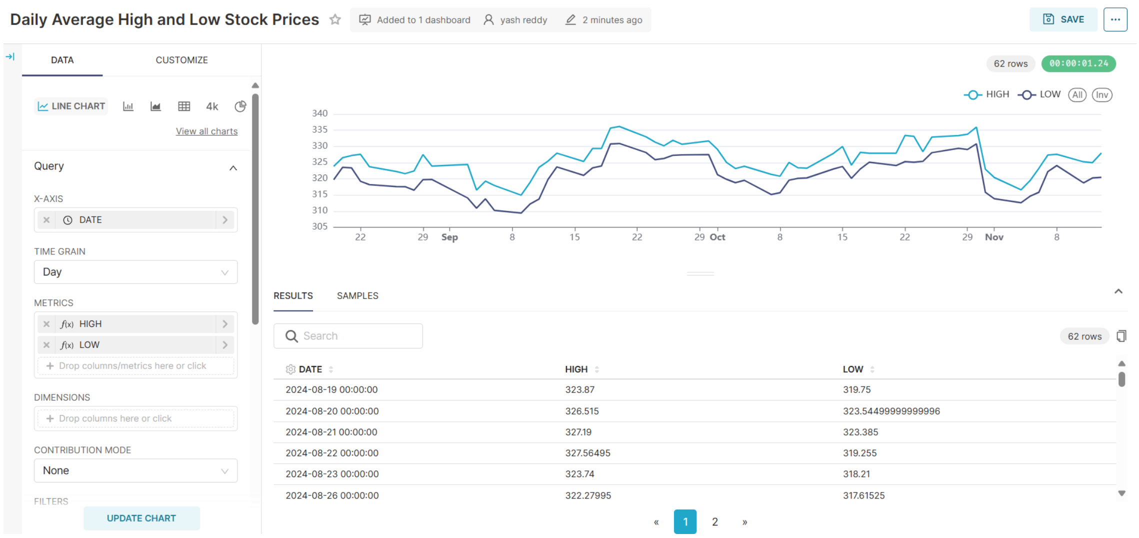


Fig. 9 Overview of the dashboard in Superset



Daily Average High and Low Stock Prices

Query

X-AXIS
DATE

TIME GRAIN
Day

METRICS
f(x) HIGH
f(x) LOW
+ Drop columns/metrics here or click

DIMENSIONS
+ Drop columns here or click

CONTRIBUTION MODE
None

FILTERS

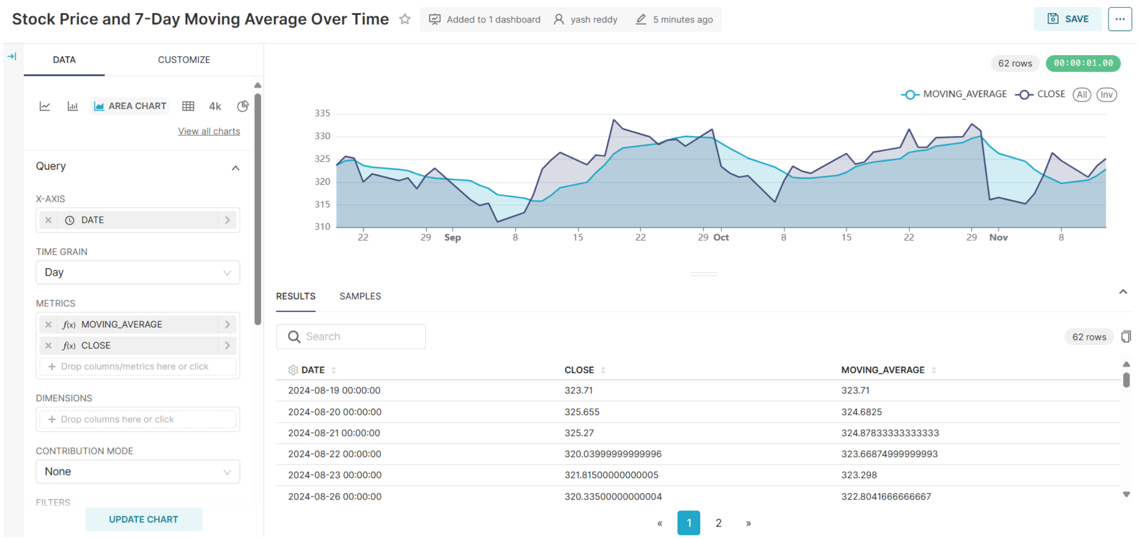
RESULTS SAMPLES

Search 62 rows

DATE	HIGH	LOW
2024-08-19 00:00:00	323.87	319.75
2024-08-20 00:00:00	326.515	323.54499999999996
2024-08-21 00:00:00	327.19	323.385
2024-08-22 00:00:00	327.56495	319.255
2024-08-23 00:00:00	323.74	318.21
2024-08-26 00:00:00	322.27995	317.61525

« 1 2 »

Fig. 10 Daily Average High and Low Stock Prices



Stock Price and 7-Day Moving Average Over Time

Query

X-AXIS
DATE

TIME GRAIN
Day

METRICS
f(x) MOVING_AVERAGE
f(x) CLOSE
+ Drop columns/metrics here or click

DIMENSIONS
+ Drop columns here or click

CONTRIBUTION MODE
None

FILTERS

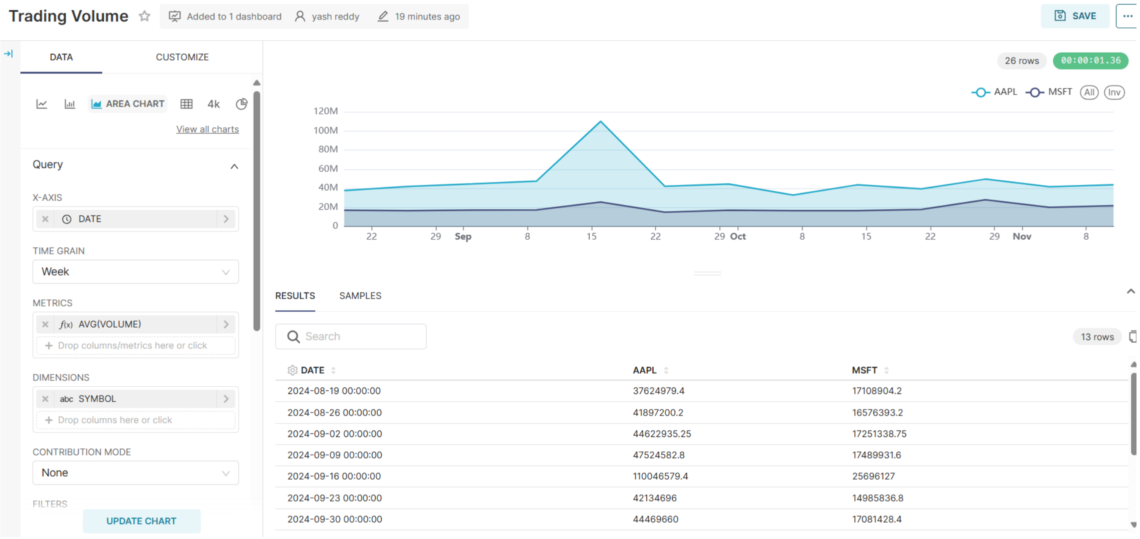
RESULTS SAMPLES

Search 62 rows

DATE	CLOSE	MOVING_AVERAGE
2024-08-19 00:00:00	323.71	323.71
2024-08-20 00:00:00	325.655	324.6825
2024-08-21 00:00:00	325.27	324.87833333333333
2024-08-22 00:00:00	320.03999999999996	323.66874999999993
2024-08-23 00:00:00	321.81500000000005	323.298
2024-08-26 00:00:00	320.33500000000004	322.8041666666667

« 1 2 »

Fig. 11 Stock price and 7-Day Moving Average Over Time



Query

X-AXIS

DATE

TIME GRAIN

Week

METRICS

AVG(VOLUME)

Drop columns/metrics here or click

DIMENSIONS

SYMBOL

Drop columns here or click

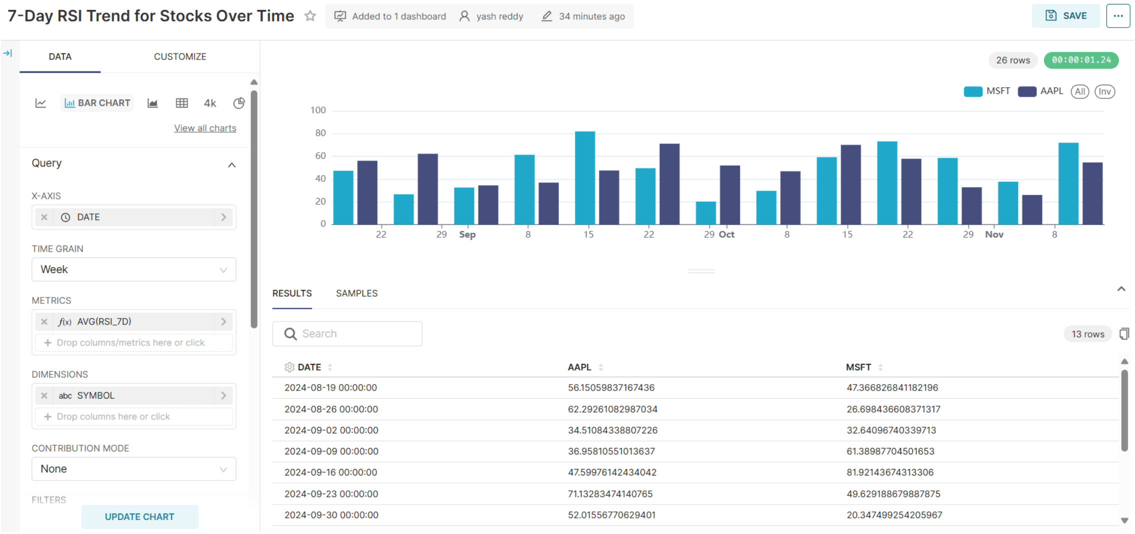
CONTRIBUTION MODE

None

FILTERS

DATE	AAPL	MSFT
2024-08-19 00:00:00	37624979.4	17108904.2
2024-08-26 00:00:00	41897200.2	16576393.2
2024-09-02 00:00:00	44622935.25	17251338.75
2024-09-09 00:00:00	47524582.8	17489931.6
2024-09-16 00:00:00	110046579.4	25696127
2024-09-23 00:00:00	42134696	14985836.8
2024-09-30 00:00:00	44469660	17081428.4

Fig. 12 Trading Volume



Query

X-AXIS

DATE

TIME GRAIN

Week

METRICS

AVG(RSI_7D)

Drop columns/metrics here or click

DIMENSIONS

SYMBOL

Drop columns here or click

CONTRIBUTION MODE

None

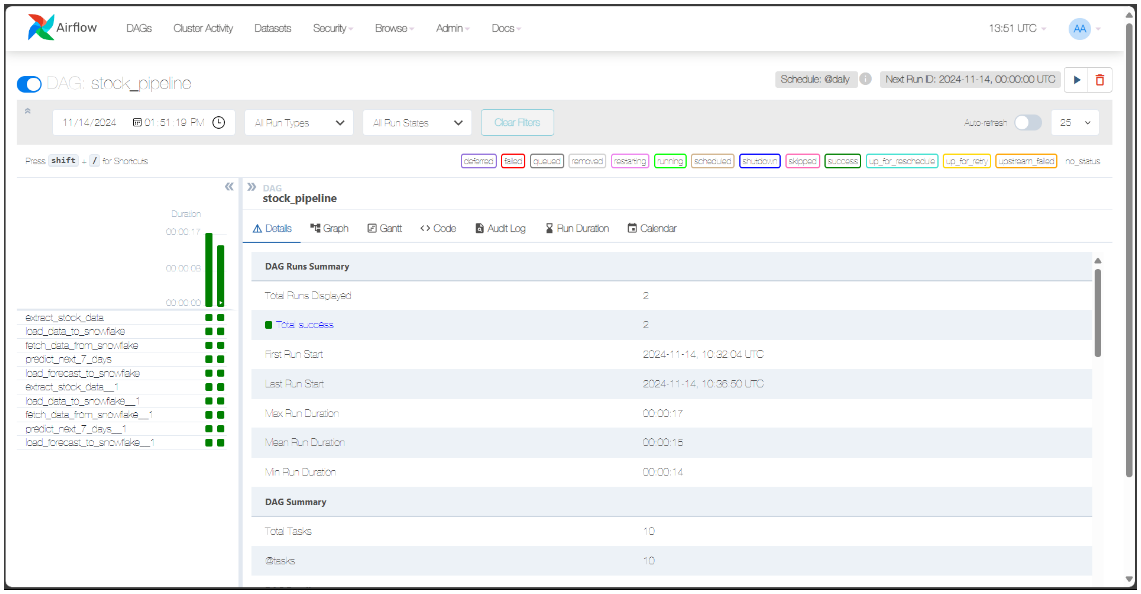
FILTERS

DATE	AAPL	MSFT
2024-08-19 00:00:00	56.15059837167436	47.366826841182196
2024-08-26 00:00:00	62.29261082987034	26.698436608371317
2024-09-02 00:00:00	34.51084338807226	32.64096740339713
2024-09-09 00:00:00	36.95810551013637	61.38987704501653
2024-09-16 00:00:00	47.59976142434042	81.92143674313306
2024-09-23 00:00:00	71.13283474140765	49.629188679887875
2024-09-30 00:00:00	52.01556770629401	20.347499254205967

Fig. 13 7 Day RSI Trend for Stocks Over Time

F. *Pipeline Orchestration*

The complete pipeline is orchestrated through Airflow, which manages the sequence of operations in a specific order. The process begins with data extraction from Alpha Vantage, followed by loading the data into Snowflake raw tables. Next, the pipeline executes dbt transformations on the loaded data. Finally, the process concludes with refreshing the visualization layer data. This methodology ensures a streamlined process from data ingestion through transformation to final visualization, maintaining data consistency and enabling automated updates of our analytical insights.



DAG: stock_pipeline

Schedule: @daily

Next Run ID: 2024-11-14, 00:00:00 UTC

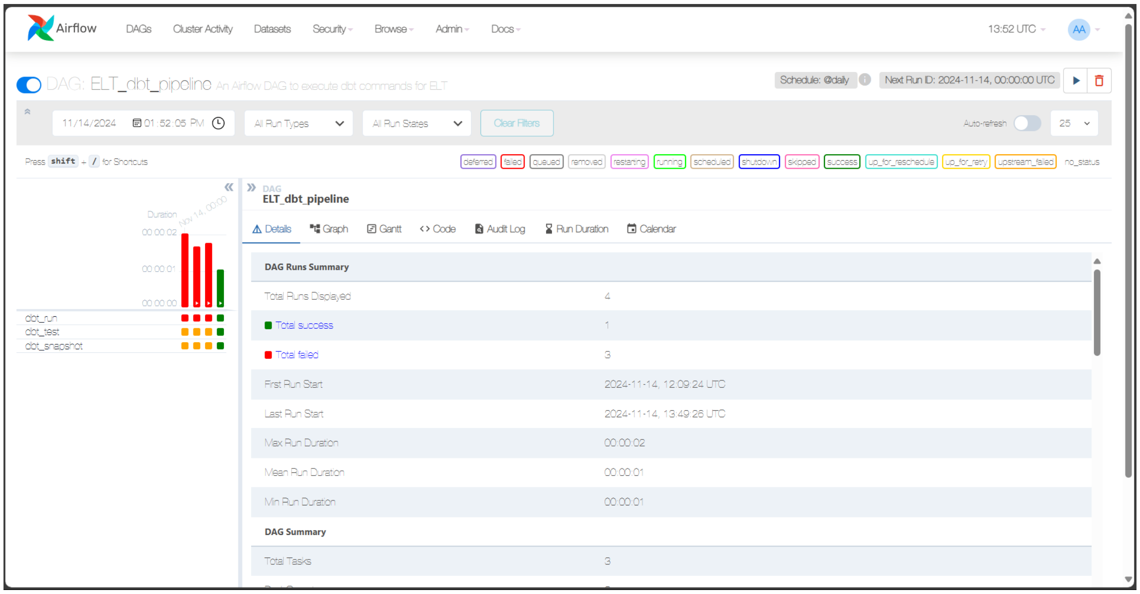
11/14/2024 01:51:19 PM

extract_stock_data
load_data_to_snowflake
fetch_data_from_snowflake
predict_next_7_days
load_forecast_to_snowflake
extract_stock_data__1
load_data_to_snowflake__1
fetch_data_from_snowflake__1
predict_next_7_days__1
load_forecast_to_snowflake__1

DAG
stock_pipeline

DAG Runs Summary	
Total Runs Displayed	2
🟩 Total success	2
First Run Start	2024-11-14, 10:32:04 UTC
Last Run Start	2024-11-14, 10:36:50 UTC
Max Run Duration	00:00:17
Mean Run Duration	00:00:15
Min Run Duration	00:00:14
DAG Summary	
Total Tasks	10
@tasks	10

Fig. 14 stock_pipeline execution in Airflow



DAG: ELT_dbt_pipeline An Airflow DAG to execute dbt commands for ELT

Schedule: @daily

Next Run ID: 2024-11-14, 00:00:00 UTC

11/14/2024 01:52:05 PM

dbt_run
dbt_test
dbt_snapshot

DAG
ELT_dbt_pipeline

DAG Runs Summary	
Total Runs Displayed	4
🟩 Total success	1
🟥 Total failed	3
First Run Start	2024-11-14, 12:09:24 UTC
Last Run Start	2024-11-14, 13:49:26 UTC
Max Run Duration	00:00:02
Mean Run Duration	00:00:01
Min Run Duration	00:00:01
DAG Summary	
Total Tasks	3

Fig. 15 ETL_dbt_pipeline execution in Airflow

V. RESULTS

This lab implemented code to realize an efficient automated data pipeline for analysis of stock market data. Apache Airflow has extracted data from the Alpha Vantage API and loaded data into raw data tables in Snowflake, maintaining data consistency on a daily refresh cycle. Also, dbt has transformed the data directly in Snowflake to create analytical tables with important key metrics such as 7 day moving averages, RSI, and price momentum. These transformed metrics then allowed for more-valuable insights into stock trends and possible trading signals. The interactive dashboards in Preset/Superset/Tableau showcased effective utilization of the BI tool. These visualizations include stock price trends, moving average analysis, RSI analysis, and volume patterns. The date filtering will be interactive, and the drill-down capability further allows deep sets of understanding about how stocks would perform over time. Airflow and dbt integrated correctly kept this pipeline consistent and fresh, refreshed data and visualization automatically.

VI. CONCLUSION

This lab will demonstrate how to create an end-to-end data pipeline for stock price analytics, focusing on ELT and visualization. With the ETL by Apache Airflow, in-warehouse transformations by dbt, and visualization by a BI tool, the pipeline was able to furnish real-time, interactive insights into the trends in the stock market. The selection of dbt for performing transformations using in Snowflake allowed for modular and scalable transformation without the need for redundant data movement. This integration ensures every part in the chain of the pipeline will keep the data consistent, accurate and timely. The dashboard derived from the this enables users to monitor and interpret key stock metrics interactively, making it easier to drive decisions with data.

VII. Output

Github Link

VII. REFERENCES

1.https://global.trocco.io/blogs/etl-vs-elt-key-differences-and-their-role-in-data-warehousing

2. https://airflowsummit.org/slides/2024/98-Exploring-DAG-Design-Patterns-in-Apache-Airflow.pdf

3.https://rhasanm.github.io/being-data-engineer/posts/data_pipelines_with_apache_airflow/

4. https://www.astronomer.io/docs/learn/dag-best-practices/

5. https://airflow.apache.org/docs/apache-airflow/stable/best-practices.html

6. https://www.datacamp.com/blog/etl-vs-elt