Lab 2: End-to-End Data Analytics using Snowflake, Airflow, dbt, and Superset

(Members: Sharan Patil and Mansi Deshmukh)

GIT- https://github.com/sharan9219790/DATAWAREHOUSE-LAB2

Abstract—

This report documents the design and implementation of a complete ELT/Analytics pipeline for stock-price analytics. Raw market data is ingested with Apache Airflow into Snowflake (RAW/Public schemas), transformed with dbt into analytics-ready models (ANALYTICS schema), and visualized in Apache Superset with multiple charts including a Line chart of moving averages, Relative Strength Index (RSI) with threshold overlays, Daily Return (%) bar chart, and a KPI card for the latest close of AAPL. The pipeline was executed end-to-end, scheduled as Airflow DAGs, and the transformation layer adheres to modular staging/intermediate/marts patterns. This paper describes environment setup, credentials management without hardcoding, key SQL models, orchestration configuration, and BI design decisions.

Index Terms—

ELT, Snowflake, Airflow, dbt, Superset, RSI, Moving Average, Daily Return, Data Engineering.

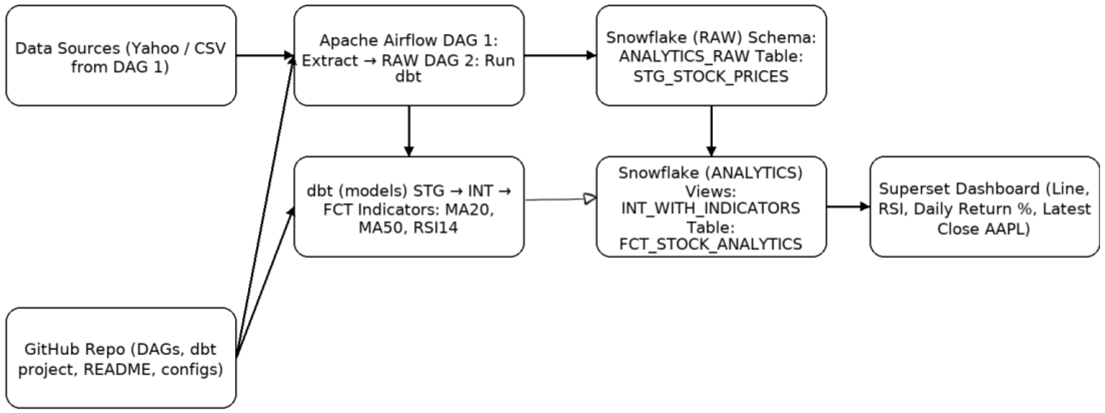
I. INTRODUCTION

The goal of Lab 2 is to build an end-to-end analytics workflow: (1) load raw data via Airflow (ETL), (2) model and transform data with dbt (ELT), and (3) visualize in a BI tool. We extended Lab 1's stock dataset and demonstrated calculations such as moving averages (MA20, MA50), RSI14, and daily returns for multiple tickers (AAPL, GOOG, MSFT). The final deliverables include running Airflow DAGs, materialized dbt models in Snowflake, and Superset dashboards. This document provides a step-by-step reproducible guide as well as rationale for each decision.

II. SYSTEM ARCHITECTURE

Figure 1 illustrates the overall architecture. Airflow orchestrates two DAGs: DAG 1 extracts/loads raw data to Snowflake (PUBLIC/RAW), and DAG 2 triggers dbt models for transformations into ANALYTICS. Superset connects to Snowflake using the Snowflake SQLAlchemy driver for dashboarding. Credentials are provided via environment variables and Airflow Connections to avoid hardcoding.

Lab 2: End-to-End Analytics Architecture (Clean)


Data Sources (Yahoo / CSV from DAG 1)
Apache Airflow DAG 1: Extract → RAW DAG 2: Run dbt
Snowflake (RAW) Schema: ANALYTICS_RAW Table: STG_STOCK_PRICES
dbt (models) STG → INT → FCT Indicators: MA20, MA50, RSI14
Snowflake (ANALYTICS) Views: INT_WITH_INDICATORS Table: FCT_STOCK_ANALYTICS
Superset Dashboard (Line, RSI, Daily Return %, Latest Close AAPL)
GitHub Repo (DAGs, dbt project, README, configs)

Flow: Sources → Airflow DAGs → Snowflake RAW → dbt → Snowflake ANALYTICS → Superset

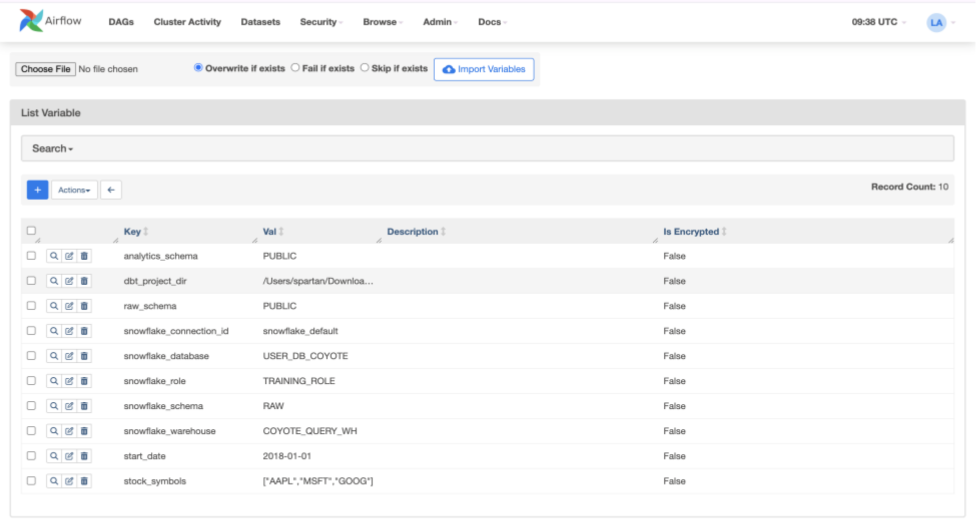
III. ENVIRONMENT & CREDENTIALS

A Python virtual environment was used for both the dbt and Superset stacks. Credentials were supplied through environment variables and reflected in dbt’s profiles.yml. The Snowflake role correction (TRAINING_ROLE) and account connectivity were validated via both dbt debug and a direct Python connector test.

Key Environment Variables (example):

```
export SNOWFLAKE_ACCOUNT=SFEDU02-LVxxxxx
export SNOWFLAKE_USER=COxxx
export SNOWFLAKE_ROLE=TRAINxxxxxx
export SNOWFLAKE_WAREHOUSE=COYOxxxxx
export SNOWFLAKE_DATABASE=USER_Dxxxxx
export SNOWFLAKE_SCHEMA=ANALYTxxx
export DBT_PROFILES_DIR=/Users/spartan/Downloads/DATA226/LAB2new/dbt
```

Airflow Variable and Connection–



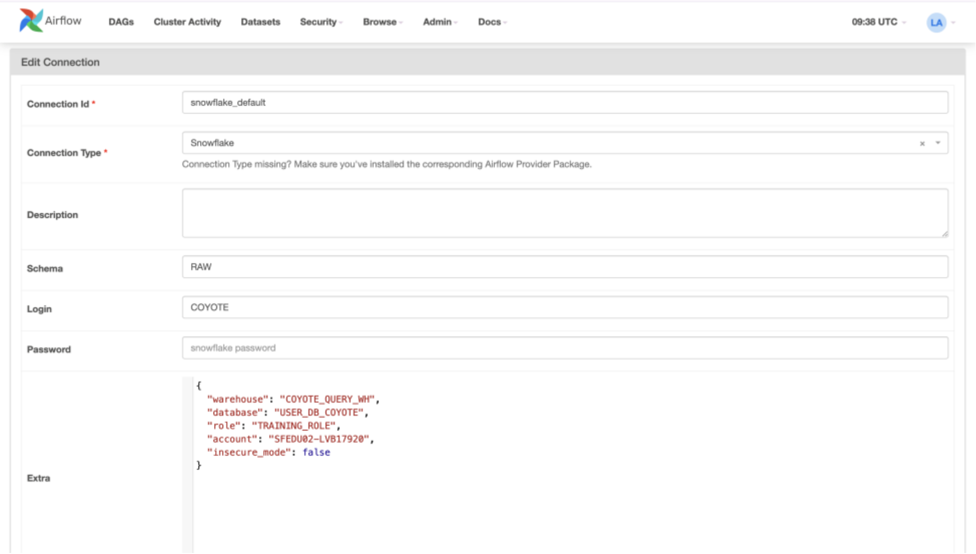
No file chosen

● Overwrite if exists ○ Fail if exists ○ Skip if exists Import Variables

List Variable

Record Count: 10

Key	Val	Description	Is Encrypted
analytics_schema	PUBLIC		False
dbt_project_dir	/Users/spartan/Downloa...		False
raw_schema	PUBLIC		False
snowflake_connection_id	snowflake_default		False
snowflake_database	USER_DB_COYOTE		False
snowflake_role	TRAINING_ROLE		False
snowflake_schema	RAW		False
snowflake_warehouse	COYOTE_QUERY_WH		False
start_date	2018-01-01		False
stock_symbols	["AAPL","MSFT","GOOG"]		False



Edit Connection

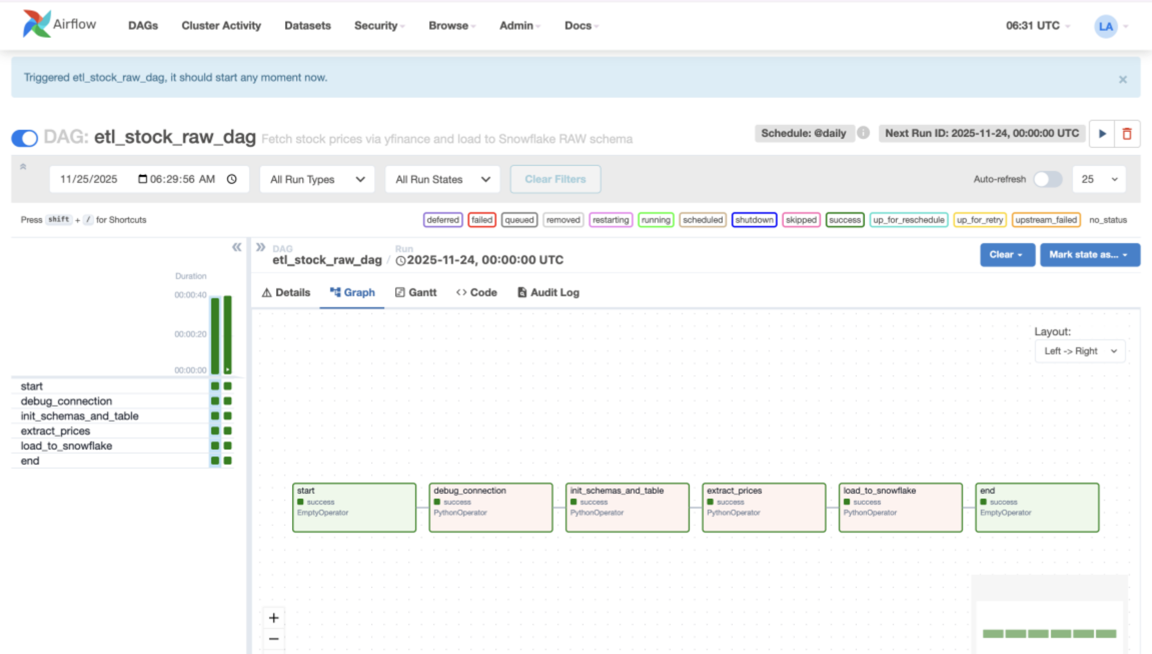
Connection Id *	snowflake_default
Connection Type *	Snowflake Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.
Description	
Schema	RAW
Login	COYOTE
Password	snowflake password
Extra	

```
{
  "warehouse": "COYOTE_QUERY_WH",
  "database": "USER_DB_COYOTE",
  "role": "TRAINING_ROLE",
  "account": "SFEDU02-LVB17920",
  "insecure_mode": false
}
```

IV. AIRFLOW ORCHESTRATION (ETL)

Two DAGs were implemented. DAG 1 loads the raw dataset into Snowflake (PUBLIC or RAW schema). DAG 2 calls dbt with the Snowflake connection configured in Airflow. Successful runs were confirmed in the Airflow UI and via Snowflake row counts. You can find all the Airflow related screenshots below.

Airflow DAG 1 Graph View –


Triggered etl_stock_raw_dag, it should start any moment now.

DAG: etl_stock_raw_dag Fetch stock prices via yfinance and load to Snowflake RAW schema

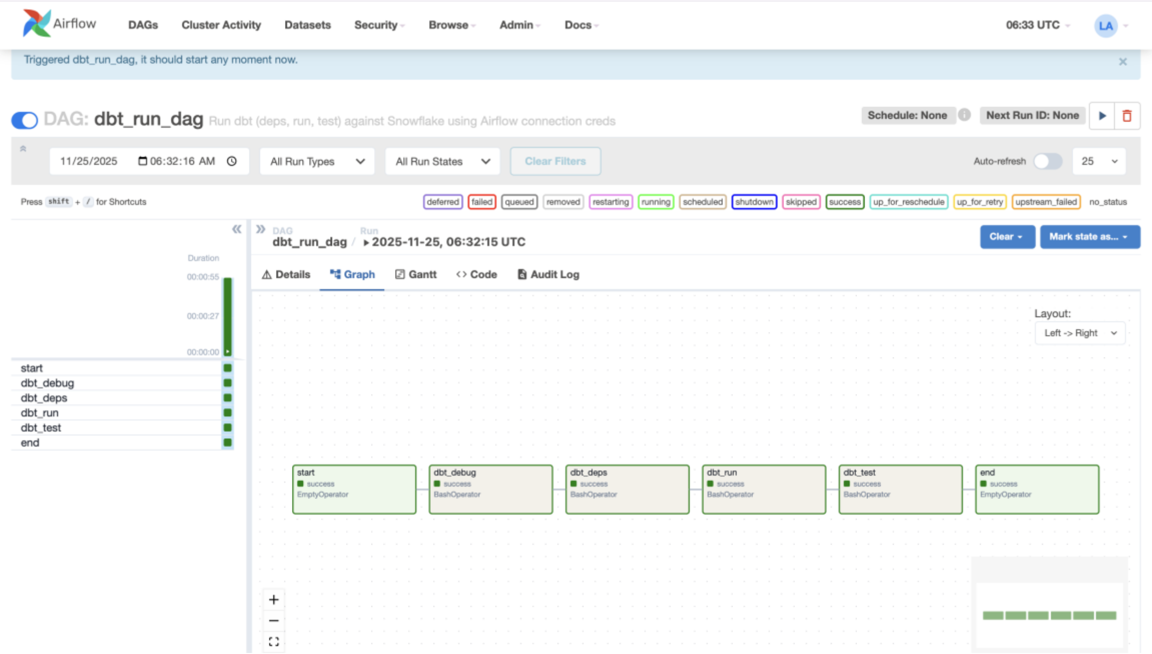
Schedule: @daily

Next Run ID: 2025-11-24, 00:00:00 UTC

11/25/2025 06:29:56 AM

DAG etl_stock_raw_dag / Run 2025-11-24, 00:00:00 UTC

Airflow DAG 2 Graph View


Triggered dbt_run_dag, it should start any moment now.

DAG: dbt_run_dag Run dbt (deps, run, test) against Snowflake using Airflow connection creds

Schedule: None

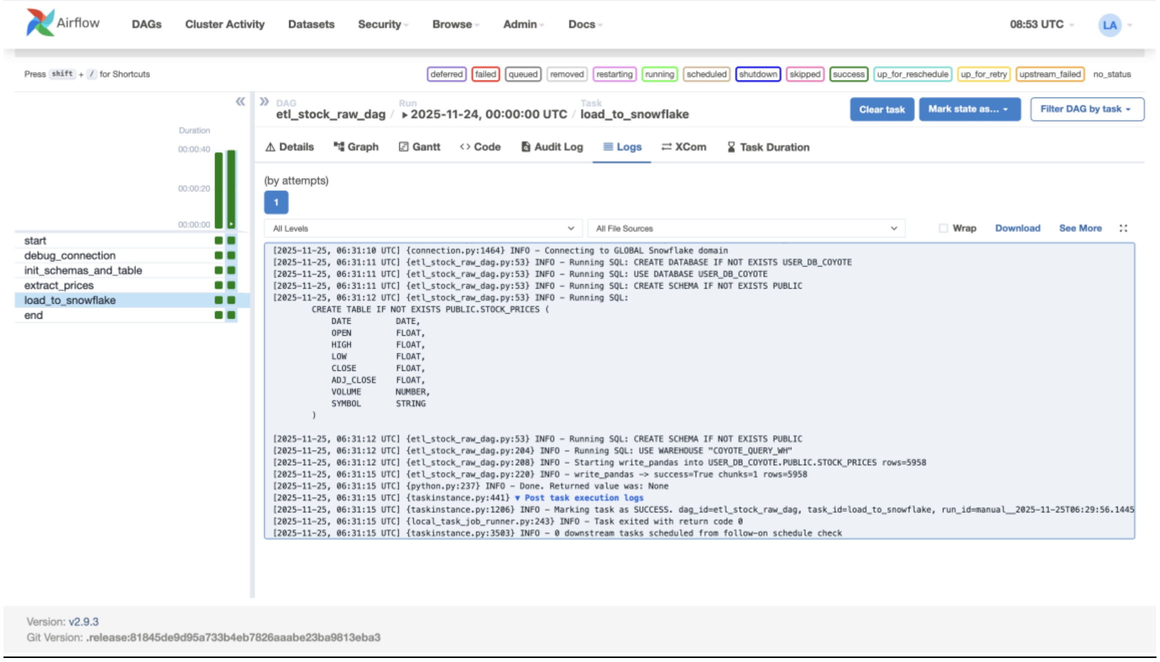
Next Run ID: None

11/25/2025 06:32:16 AM

DAG dbt_run_dag / Run 2025-11-25, 06:32:15 UTC

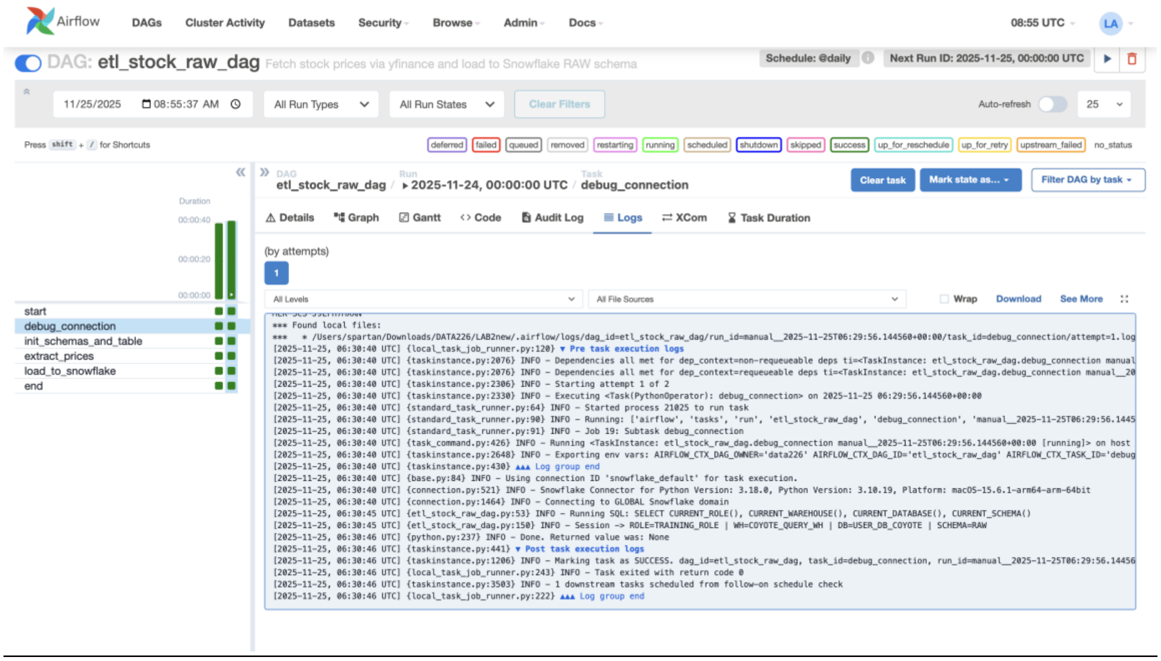
Airflow Task Logs/Success for both DAGs –

DAG1 logs

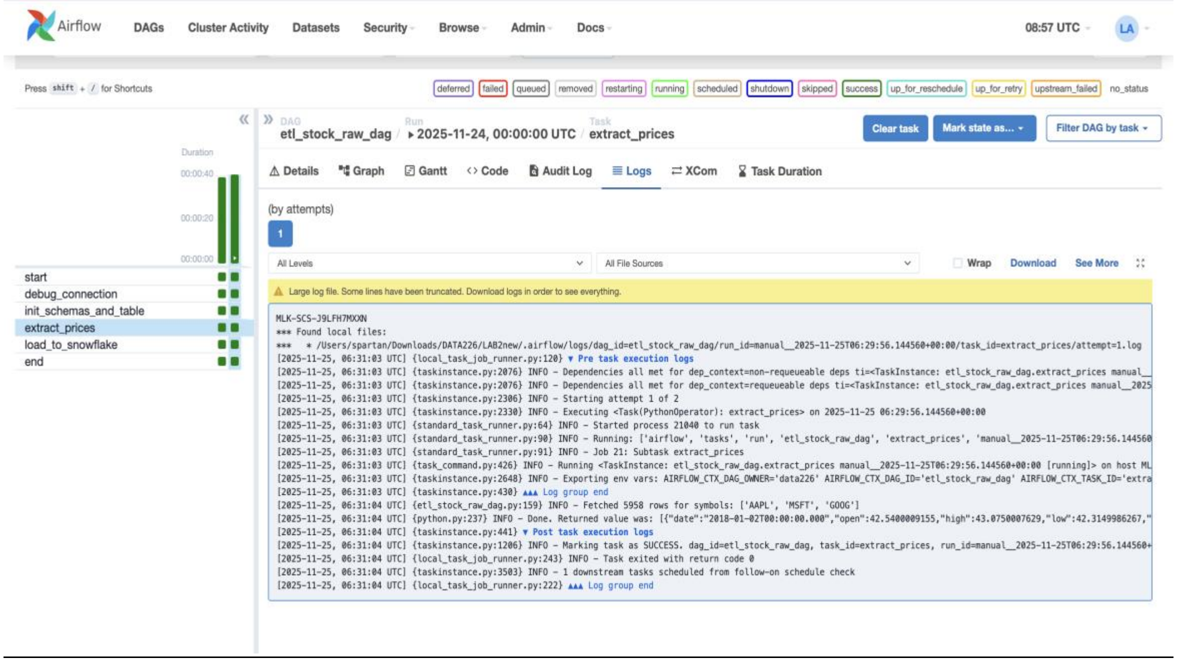


```
[2025-11-25, 06:31:10 UTC] {connection.py:1464} INFO - Connecting to GLOBAL Snowflake domain
[2025-11-25, 06:31:11 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL: CREATE DATABASE IF NOT EXISTS USER_DB_COYOTE
[2025-11-25, 06:31:11 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL: USE DATABASE USER_DB_COYOTE
[2025-11-25, 06:31:11 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL: CREATE SCHEMA IF NOT EXISTS PUBLIC
[2025-11-25, 06:31:12 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL:
        CREATE TABLE IF NOT EXISTS PUBLIC.STOCK_PRICES (
            DATE         DATE,
            OPEN         FLOAT,
            HIGH         FLOAT,
            LOW          FLOAT,
            CLOSE        FLOAT,
            ADJ_CLOSE    FLOAT,
            VOLUME       NUMBER,
            SYMBOL       STRING
        )

[2025-11-25, 06:31:12 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL: CREATE SCHEMA IF NOT EXISTS PUBLIC
[2025-11-25, 06:31:12 UTC] {etl_stock_raw_dag.py:204} INFO - Running SQL: USE WAREHOUSE "COYOTE_QUERY_WH"
[2025-11-25, 06:31:12 UTC] {etl_stock_raw_dag.py:208} INFO - Starting write_pandas into USER_DB_COYOTE.PUBLIC.STOCK_PRICES rows=5958
[2025-11-25, 06:31:15 UTC] {etl_stock_raw_dag.py:220} INFO - write_pandas -> success=True chunks=1 rows=5958
[2025-11-25, 06:31:15 UTC] {python.py:237} INFO - Done. Returned value was: None
[2025-11-25, 06:31:15 UTC] {taskinstance.py:441} ▼ Post task execution logs
[2025-11-25, 06:31:15 UTC] {taskinstance.py:1206} INFO - Marking task as SUCCESS. dag_id=etl_stock_raw_dag, task_id=load_to_snowflake, run_id=manual__2025-11-25T06:29:56.1445
[2025-11-25, 06:31:15 UTC] {local_task_job_runner.py:243} INFO - Task exited with return code 0
[2025-11-25, 06:31:15 UTC] {taskinstance.py:3503} INFO - 0 downstream tasks scheduled from follow-on schedule check
```

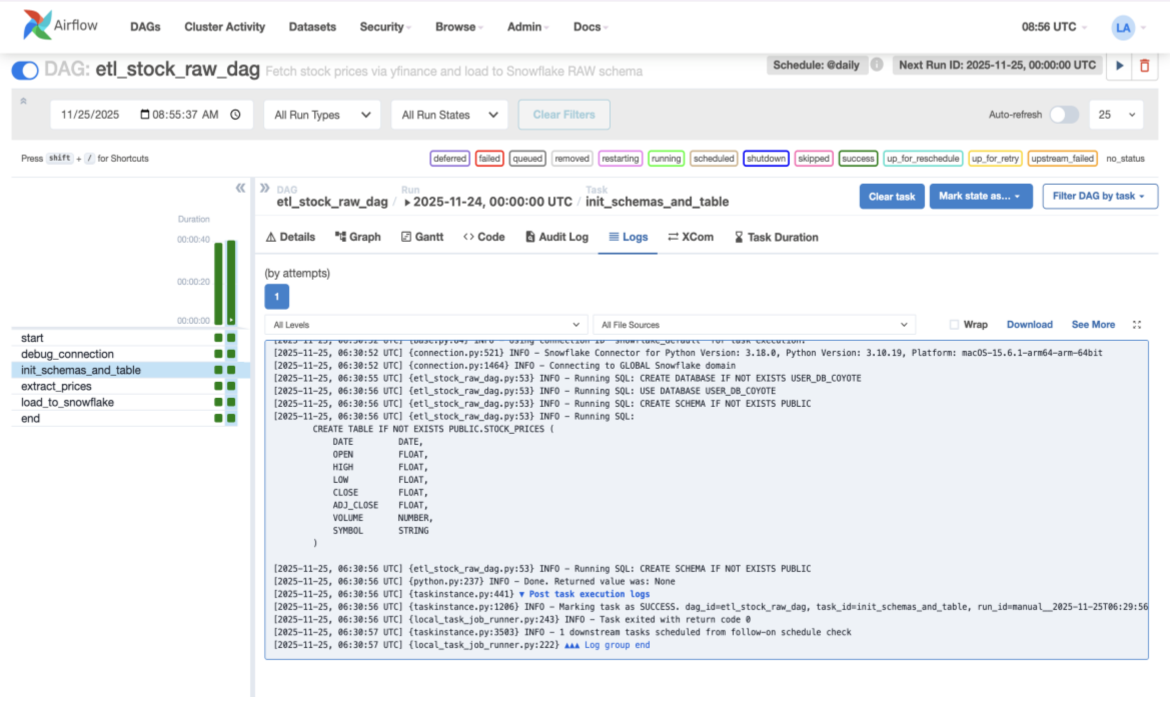


```
*** Found local files:
***   * /Users/spartan/Downloads/DATA226/LAB2new/.airflow/logs/dag_id=etl_stock_raw_dag/run_id=manual__2025-11-25T06:29:56.144560+00:00/task_id=debug_connection/attempt=1.log
[2025-11-25, 06:30:40 UTC] {local_task_job_runner.py:120} ▼ Pre task execution logs
[2025-11-25, 06:30:40 UTC] {taskinstance.py:2076} INFO - Dependencies all met for dep_context=non-requeueable deps ti=<TaskInstance: etl_stock_raw_dag.debug_connection manual
[2025-11-25, 06:30:40 UTC] {taskinstance.py:2076} INFO - Dependencies all met for dep_context=requeueable deps ti=<TaskInstance: etl_stock_raw_dag.debug_connection manual__20
[2025-11-25, 06:30:40 UTC] {taskinstance.py:2306} INFO - Starting attempt 1 of 2
[2025-11-25, 06:30:40 UTC] {taskinstance.py:2330} INFO - Executing <Task(PythonOperator): debug_connection> on 2025-11-25 06:29:56.144560+00:00
[2025-11-25, 06:30:40 UTC] {standard_task_runner.py:64} INFO - Started process 21025 to run task
[2025-11-25, 06:30:40 UTC] {standard_task_runner.py:90} INFO - Running: ['airflow', 'tasks', 'run', 'etl_stock_raw_dag', 'debug_connection', 'manual__2025-11-25T06:29:56.1445
[2025-11-25, 06:30:40 UTC] {standard_task_runner.py:91} INFO - Job 19: Subtask debug_connection
[2025-11-25, 06:30:40 UTC] {task_command.py:426} INFO - Running <TaskInstance: etl_stock_raw_dag.debug_connection manual__2025-11-25T06:29:56.144560+00:00 [running]> on host
[2025-11-25, 06:30:40 UTC] {taskinstance.py:2648} INFO - Exporting env vars: AIRFLOW_CTX_DAG_OWNER='data226' AIRFLOW_CTX_DAG_ID='etl_stock_raw_dag' AIRFLOW_CTX_TASK_ID='debug
[2025-11-25, 06:30:40 UTC] {taskinstance.py:430} ▲▲▲ Log group end
[2025-11-25, 06:30:40 UTC] {base.py:84} INFO - Using connection ID 'snowflake_default' for task execution.
[2025-11-25, 06:30:40 UTC] {connection.py:521} INFO - Snowflake Connector for Python Version: 3.18.0, Python Version: 3.10.19, Platform: macOS-15.6.1-arm64-arm-64bit
[2025-11-25, 06:30:40 UTC] {connection.py:1464} INFO - Connecting to GLOBAL Snowflake domain
[2025-11-25, 06:30:45 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL: SELECT CURRENT_ROLE(), CURRENT_WAREHOUSE(), CURRENT_DATABASE(), CURRENT_SCHEMA()
[2025-11-25, 06:30:45 UTC] {etl_stock_raw_dag.py:150} INFO - Session -> ROLE=TRAINING_ROLE | WH=COYOTE_QUERY_WH | DB=USER_DB_COYOTE | SCHEMA=RAW
[2025-11-25, 06:30:46 UTC] {python.py:237} INFO - Done. Returned value was: None
[2025-11-25, 06:30:46 UTC] {taskinstance.py:441} ▼ Post task execution logs
[2025-11-25, 06:30:46 UTC] {taskinstance.py:1206} INFO - Marking task as SUCCESS. dag_id=etl_stock_raw_dag, task_id=debug_connection, run_id=manual__2025-11-25T06:29:56.14456
[2025-11-25, 06:30:46 UTC] {local_task_job_runner.py:243} INFO - Task exited with return code 0
[2025-11-25, 06:30:46 UTC] {taskinstance.py:3503} INFO - 1 downstream tasks scheduled from follow-on schedule check
[2025-11-25, 06:30:46 UTC] {local_task_job_runner.py:222} ▲▲▲ Log group end
```



Large log file. Some lines have been truncated. Download logs in order to see everything.

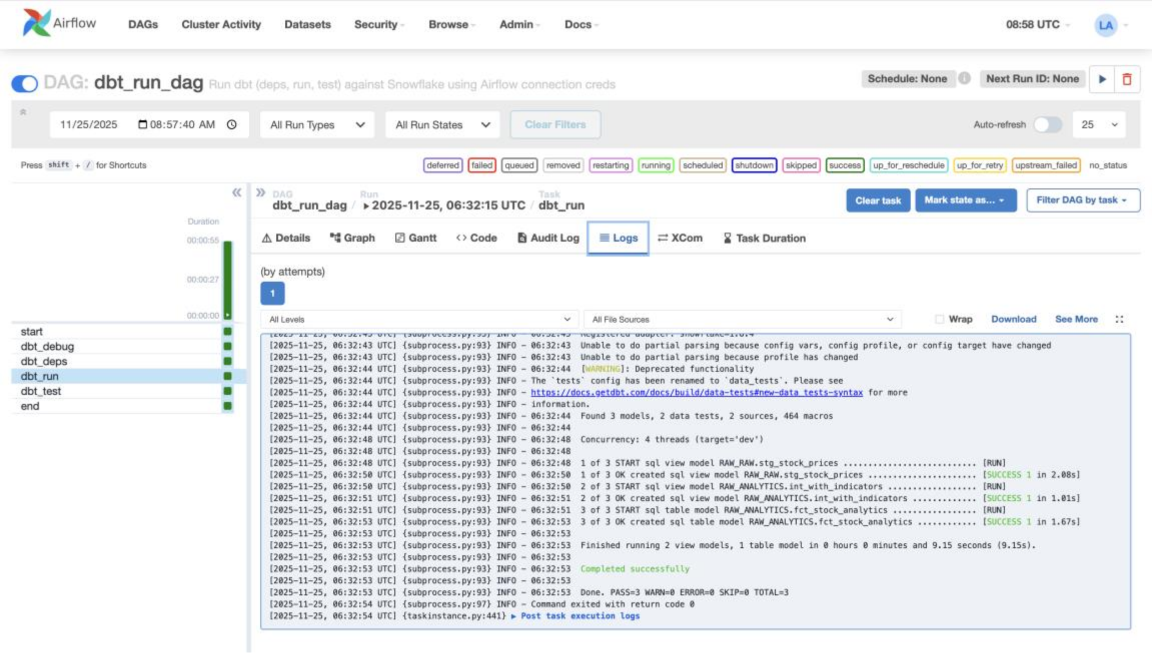
```
MLK-SCS-J9LFH7MXXN
*** Found local files:
***   * /Users/spartan/Downloads/DATA226/LAB2new/.airflow/logs/dag_id=etl_stock_raw_dag/run_id=manual__2025-11-25T06:29:56.144560+00:00/task_id=extract_prices/attempt=1.log
[2025-11-25, 06:31:03 UTC] {local_task_job_runner.py:120} ▼ Pre task execution logs
[2025-11-25, 06:31:03 UTC] {taskinstance.py:2076} INFO - Dependencies all met for dep_context=non-requeueable deps ti=<TaskInstance: etl_stock_raw_dag.extract_prices manual__
[2025-11-25, 06:31:03 UTC] {taskinstance.py:2076} INFO - Dependencies all met for dep_context=requeueable deps ti=<TaskInstance: etl_stock_raw_dag.extract_prices manual__2025
[2025-11-25, 06:31:03 UTC] {taskinstance.py:2306} INFO - Starting attempt 1 of 2
[2025-11-25, 06:31:03 UTC] {taskinstance.py:2330} INFO - Executing <Task(PythonOperator): extract_prices> on 2025-11-25 06:29:56.144560+00:00
[2025-11-25, 06:31:03 UTC] {standard_task_runner.py:64} INFO - Started process 21040 to run task
[2025-11-25, 06:31:03 UTC] {standard_task_runner.py:90} INFO - Running: ['airflow', 'tasks', 'run', 'etl_stock_raw_dag', 'extract_prices', 'manual__2025-11-25T06:29:56.144560
[2025-11-25, 06:31:03 UTC] {standard_task_runner.py:91} INFO - Job 21: Subtask extract_prices
[2025-11-25, 06:31:03 UTC] {task_command.py:426} INFO - Running <TaskInstance: etl_stock_raw_dag.extract_prices manual__2025-11-25T06:29:56.144560+00:00 [running]> on host ML
[2025-11-25, 06:31:03 UTC] {taskinstance.py:2648} INFO - Exporting env vars: AIRFLOW_CTX_DAG_OWNER='data226' AIRFLOW_CTX_DAG_ID='etl_stock_raw_dag' AIRFLOW_CTX_TASK_ID='extra
[2025-11-25, 06:31:03 UTC] {taskinstance.py:430} ▲▲▲ Log group end
[2025-11-25, 06:31:04 UTC] {etl_stock_raw_dag.py:159} INFO - Fetched 5958 rows for symbols: ['AAPL', 'MSFT', 'GOOG']
[2025-11-25, 06:31:04 UTC] {python.py:237} INFO - Done. Returned value was: [{"date":"2018-01-02T00:00:00.000","open":42.5400009155,"high":43.0750007629,"low":42.3149986267,"
[2025-11-25, 06:31:04 UTC] {taskinstance.py:441} ▼ Post task execution logs
[2025-11-25, 06:31:04 UTC] {taskinstance.py:1206} INFO - Marking task as SUCCESS. dag_id=etl_stock_raw_dag, task_id=extract_prices, run_id=manual__2025-11-25T06:29:56.144560+
[2025-11-25, 06:31:04 UTC] {local_task_job_runner.py:243} INFO - Task exited with return code 0
[2025-11-25, 06:31:04 UTC] {taskinstance.py:3503} INFO - 1 downstream tasks scheduled from follow-on schedule check
[2025-11-25, 06:31:04 UTC] {local_task_job_runner.py:222} ▲▲▲ Log group end
```



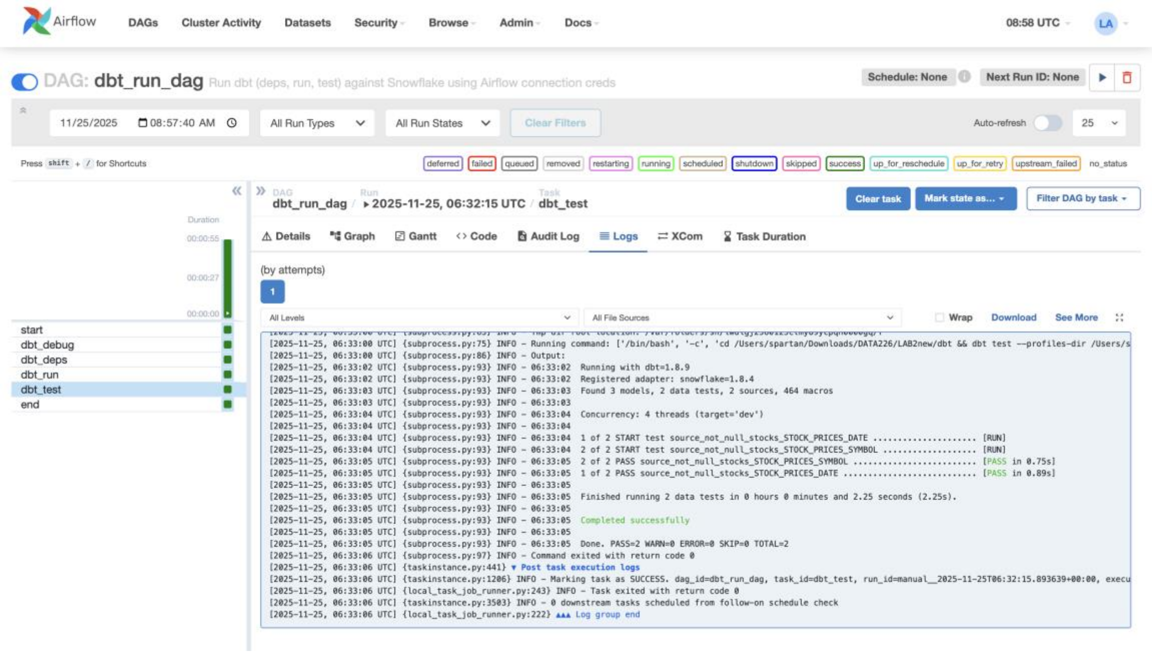
```
[2025-11-25, 06:30:52 UTC] {connection.py:521} INFO - Snowflake Connector for Python Version: 3.18.0, Python Version: 3.10.19, Platform: macOS-15.6.1-arm64-arm-64bit
[2025-11-25, 06:30:52 UTC] {connection.py:1464} INFO - Connecting to GLOBAL Snowflake domain
[2025-11-25, 06:30:55 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL: CREATE DATABASE IF NOT EXISTS USER_DB_COYOTE
[2025-11-25, 06:30:56 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL: USE DATABASE USER_DB_COYOTE
[2025-11-25, 06:30:56 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL: CREATE SCHEMA IF NOT EXISTS PUBLIC
[2025-11-25, 06:30:56 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL:
        CREATE TABLE IF NOT EXISTS PUBLIC.STOCK_PRICES (
            DATE         DATE,
            OPEN         FLOAT,
            HIGH         FLOAT,
            LOW          FLOAT,
            CLOSE        FLOAT,
            ADJ_CLOSE    FLOAT,
            VOLUME       NUMBER,
            SYMBOL       STRING
        )

[2025-11-25, 06:30:56 UTC] {etl_stock_raw_dag.py:53} INFO - Running SQL: CREATE SCHEMA IF NOT EXISTS PUBLIC
[2025-11-25, 06:30:56 UTC] {python.py:237} INFO - Done. Returned value was: None
[2025-11-25, 06:30:56 UTC] {taskinstance.py:441} ▼ Post task execution logs
[2025-11-25, 06:30:56 UTC] {taskinstance.py:1206} INFO - Marking task as SUCCESS. dag_id=etl_stock_raw_dag, task_id=init_schemas_and_table, run_id=manual__2025-11-25T06:29:56
[2025-11-25, 06:30:56 UTC] {local_task_job_runner.py:243} INFO - Task exited with return code 0
[2025-11-25, 06:30:57 UTC] {taskinstance.py:3503} INFO - 1 downstream tasks scheduled from follow-on schedule check
[2025-11-25, 06:30:57 UTC] {local_task_job_runner.py:222} ▲▲▲ Log group end
```

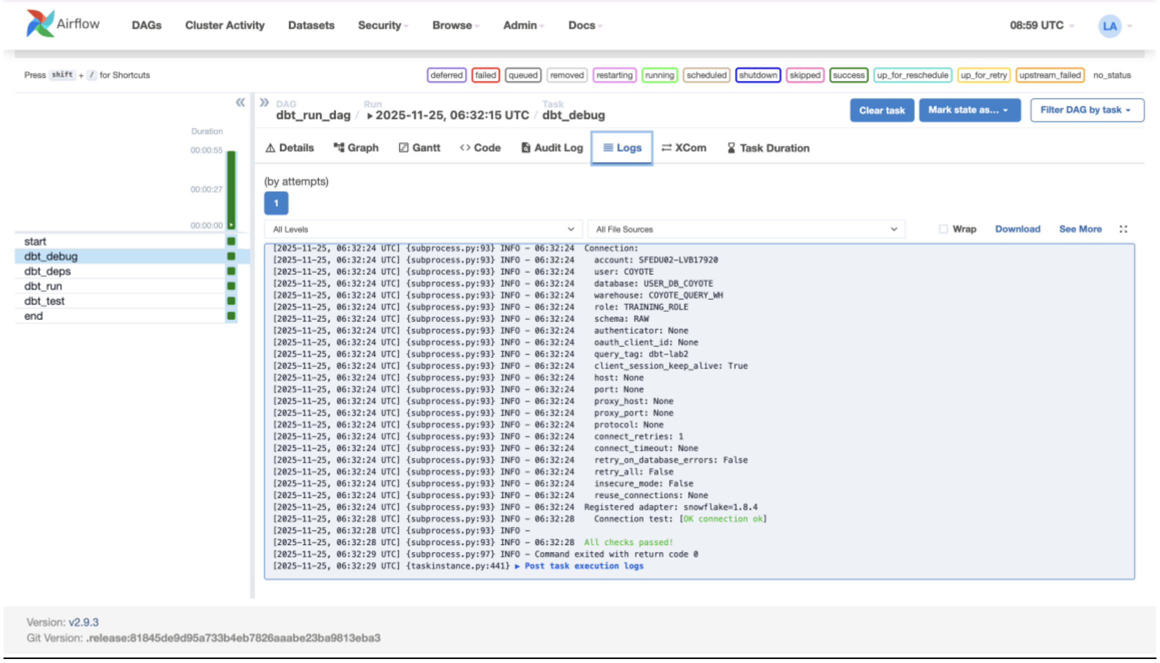
DAG2 logs-



```
[2025-11-25, 06:32:43 UTC] {subprocess.py:93} INFO - 06:32:43  Registered adapter: snowflake=1.8.4
[2025-11-25, 06:32:43 UTC] {subprocess.py:93} INFO - 06:32:43  Unable to do partial parsing because config vars, config profile, or config target have changed
[2025-11-25, 06:32:43 UTC] {subprocess.py:93} INFO - 06:32:43  Unable to do partial parsing because profile has changed
[2025-11-25, 06:32:44 UTC] {subprocess.py:93} INFO - 06:32:44  [WARNING]: Deprecated functionality
[2025-11-25, 06:32:44 UTC] {subprocess.py:93} INFO - The `tests` config has been renamed to `data_tests`. Please see
[2025-11-25, 06:32:44 UTC] {subprocess.py:93} INFO - https://docs.getdbt.com/docs/build/data-tests#new-data_tests-syntax for more
[2025-11-25, 06:32:44 UTC] {subprocess.py:93} INFO - information.
[2025-11-25, 06:32:44 UTC] {subprocess.py:93} INFO - 06:32:44  Found 3 models, 2 data tests, 2 sources, 464 macros
[2025-11-25, 06:32:44 UTC] {subprocess.py:93} INFO - 06:32:44
[2025-11-25, 06:32:48 UTC] {subprocess.py:93} INFO - 06:32:48  Concurrency: 4 threads (target='dev')
[2025-11-25, 06:32:48 UTC] {subprocess.py:93} INFO - 06:32:48
[2025-11-25, 06:32:48 UTC] {subprocess.py:93} INFO - 06:32:48  1 of 3 START sql view model RAW_RAW.stg_stock_prices .......................... [RUN]
[2025-11-25, 06:32:50 UTC] {subprocess.py:93} INFO - 06:32:50  1 of 3 OK created sql view model RAW_RAW.stg_stock_prices ..................... [SUCCESS 1 in 2.08s]
[2025-11-25, 06:32:50 UTC] {subprocess.py:93} INFO - 06:32:50  2 of 3 START sql view model RAW_ANALYTICS.int_with_indicators ................. [RUN]
[2025-11-25, 06:32:51 UTC] {subprocess.py:93} INFO - 06:32:51  2 of 3 OK created sql view model RAW_ANALYTICS.int_with_indicators ............ [SUCCESS 1 in 1.01s]
[2025-11-25, 06:32:51 UTC] {subprocess.py:93} INFO - 06:32:51  3 of 3 START sql table model RAW_ANALYTICS.fct_stock_analytics ................ [RUN]
[2025-11-25, 06:32:53 UTC] {subprocess.py:93} INFO - 06:32:53  3 of 3 OK created sql table model RAW_ANALYTICS.fct_stock_analytics ........... [SUCCESS 1 in 1.67s]
[2025-11-25, 06:32:53 UTC] {subprocess.py:93} INFO - 06:32:53
[2025-11-25, 06:32:53 UTC] {subprocess.py:93} INFO - 06:32:53  Finished running 2 view models, 1 table model in 0 hours 0 minutes and 9.15 seconds (9.15s).
[2025-11-25, 06:32:53 UTC] {subprocess.py:93} INFO - 06:32:53
[2025-11-25, 06:32:53 UTC] {subprocess.py:93} INFO - 06:32:53  Completed successfully
[2025-11-25, 06:32:53 UTC] {subprocess.py:93} INFO - 06:32:53
[2025-11-25, 06:32:53 UTC] {subprocess.py:93} INFO - 06:32:53  Done. PASS=3 WARN=0 ERROR=0 SKIP=0 TOTAL=3
[2025-11-25, 06:32:54 UTC] {subprocess.py:97} INFO - Command exited with return code 0
[2025-11-25, 06:32:54 UTC] {taskinstance.py:441} ▶ Post task execution logs
```



```
[2025-11-25, 06:33:00 UTC] {subprocess.py:75} INFO - Running command: ['/bin/bash', '-c', 'cd /Users/spartan/Downloads/DATA226/LAB2new/dbt && dbt test --profiles-dir /Users/s
[2025-11-25, 06:33:00 UTC] {subprocess.py:86} INFO - Output:
[2025-11-25, 06:33:02 UTC] {subprocess.py:93} INFO - 06:33:02  Running with dbt=1.8.9
[2025-11-25, 06:33:02 UTC] {subprocess.py:93} INFO - 06:33:02  Registered adapter: snowflake=1.8.4
[2025-11-25, 06:33:03 UTC] {subprocess.py:93} INFO - 06:33:03  Found 3 models, 2 data tests, 2 sources, 464 macros
[2025-11-25, 06:33:03 UTC] {subprocess.py:93} INFO - 06:33:03
[2025-11-25, 06:33:04 UTC] {subprocess.py:93} INFO - 06:33:04  Concurrency: 4 threads (target='dev')
[2025-11-25, 06:33:04 UTC] {subprocess.py:93} INFO - 06:33:04
[2025-11-25, 06:33:04 UTC] {subprocess.py:93} INFO - 06:33:04  1 of 2 START test source_not_null_stocks_STOCK_PRICES_DATE ..................... [RUN]
[2025-11-25, 06:33:04 UTC] {subprocess.py:93} INFO - 06:33:04  2 of 2 START test source_not_null_stocks_STOCK_PRICES_SYMBOL ................... [RUN]
[2025-11-25, 06:33:05 UTC] {subprocess.py:93} INFO - 06:33:05  2 of 2 PASS source_not_null_stocks_STOCK_PRICES_SYMBOL ......................... [PASS in 0.75s]
[2025-11-25, 06:33:05 UTC] {subprocess.py:93} INFO - 06:33:05  1 of 2 PASS source_not_null_stocks_STOCK_PRICES_DATE ........................... [PASS in 0.89s]
[2025-11-25, 06:33:05 UTC] {subprocess.py:93} INFO - 06:33:05
[2025-11-25, 06:33:05 UTC] {subprocess.py:93} INFO - 06:33:05  Finished running 2 data tests in 0 hours 0 minutes and 2.25 seconds (2.25s).
[2025-11-25, 06:33:05 UTC] {subprocess.py:93} INFO - 06:33:05
[2025-11-25, 06:33:05 UTC] {subprocess.py:93} INFO - 06:33:05  Completed successfully
[2025-11-25, 06:33:05 UTC] {subprocess.py:93} INFO - 06:33:05
[2025-11-25, 06:33:05 UTC] {subprocess.py:93} INFO - 06:33:05  Done. PASS=2 WARN=0 ERROR=0 SKIP=0 TOTAL=2
[2025-11-25, 06:33:06 UTC] {subprocess.py:97} INFO - Command exited with return code 0
[2025-11-25, 06:33:06 UTC] {taskinstance.py:441} ▼ Post task execution logs
[2025-11-25, 06:33:06 UTC] {taskinstance.py:1206} INFO - Marking task as SUCCESS. dag_id=dbt_run_dag, task_id=dbt_test, run_id=manual__2025-11-25T06:32:15.893639+00:00, execu
[2025-11-25, 06:33:06 UTC] {local_task_job_runner.py:243} INFO - Task exited with return code 0
[2025-11-25, 06:33:06 UTC] {taskinstance.py:3503} INFO - 0 downstream tasks scheduled from follow-on schedule check
[2025-11-25, 06:33:06 UTC] {local_task_job_runner.py:222} ▲▲▲ Log group end
```



```
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24  Connection:
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    account: SFEDU02-LVB17920
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    user: COYOTE
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    database: USER_DB_COYOTE
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    warehouse: COYOTE_QUERY_WH
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    role: TRAINING_ROLE
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    schema: RAW
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    authenticator: None
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    oauth_client_id: None
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    query_tag: dbt-lab2
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    client_session_keep_alive: True
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    host: None
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    port: None
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    proxy_host: None
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    proxy_port: None
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    protocol: None
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    connect_retries: 1
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    connect_timeout: None
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    retry_on_database_errors: False
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    retry_all: False
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    insecure_mode: False
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24    reuse_connections: None
[2025-11-25, 06:32:24 UTC] {subprocess.py:93} INFO - 06:32:24  Registered adapter: snowflake=1.8.4
[2025-11-25, 06:32:28 UTC] {subprocess.py:93} INFO - 06:32:28    Connection test: [OK connection ok]
[2025-11-25, 06:32:28 UTC] {subprocess.py:93} INFO -
[2025-11-25, 06:32:28 UTC] {subprocess.py:93} INFO - 06:32:28  All checks passed!
[2025-11-25, 06:32:29 UTC] {subprocess.py:97} INFO - Command exited with return code 0
[2025-11-25, 06:32:29 UTC] {taskinstance.py:441} ▶ Post task execution logs
```

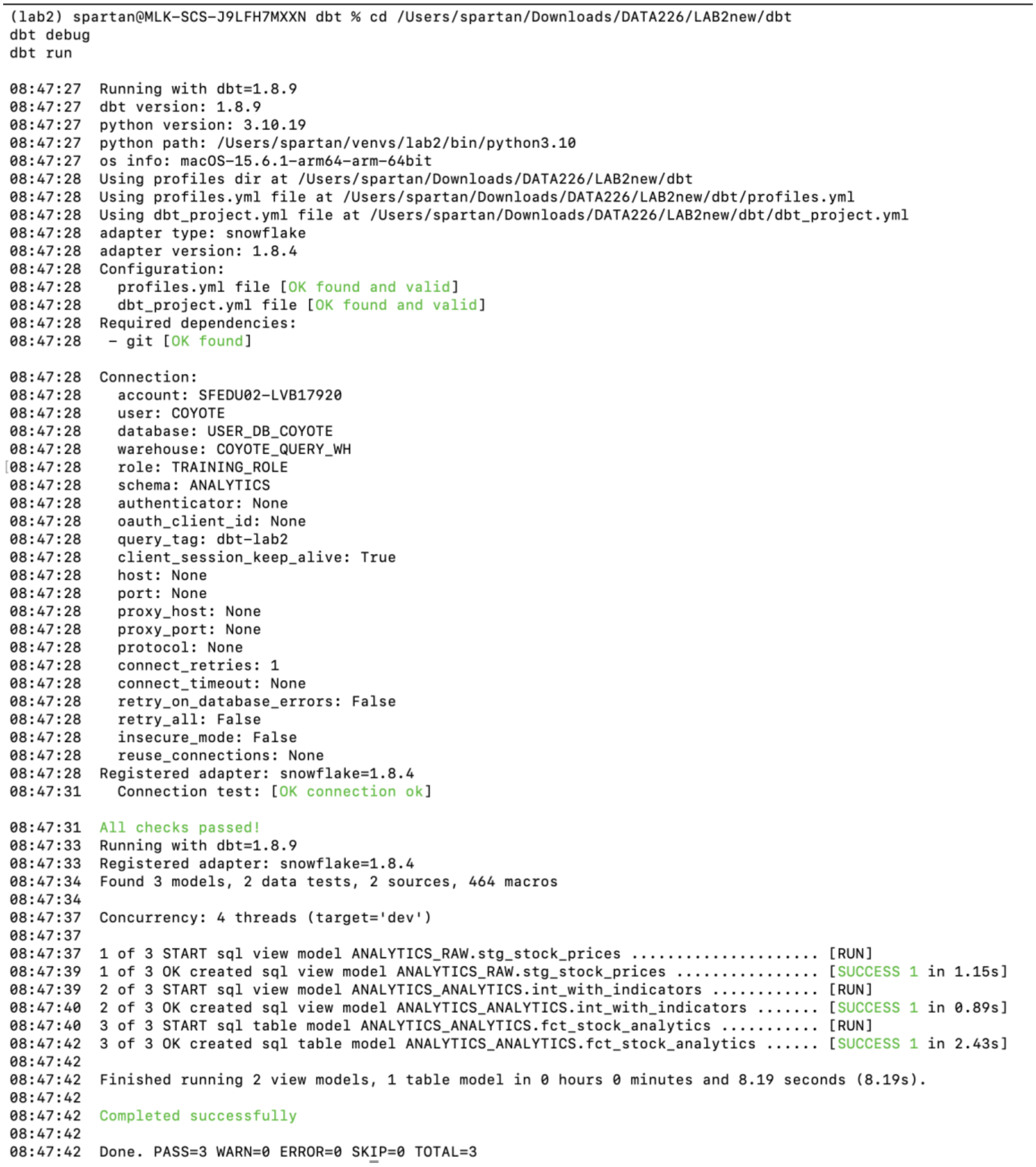
V. DBT MODELING (ELT)

We used a three-layer structure: staging (RAW → stg_stock_prices view), intermediate (int_with_indicators), and marts (fct_stock_analytics table). The dbt_profile and project were set to use environment variables, avoiding credential hardcoding. Successful execution produced:
• ANALYTICS_RAW.STG_STOCK_PRICES (view)
• ANALYTICS_ANALYTICS.INT_WITH_INDICATORS (view)
• ANALYTICS_ANALYTICS.FCT_STOCK_ANALYTICS (table)

Sample commands:

```
dbt debug
dbt run
dbt test
```

<u>Terminal output for dbt debug/run – (next page)</u>



```
(lab2) spartan@MLK-SCS-J9LFH7MXXN dbt % cd /Users/spartan/Downloads/DATA226/LAB2new/dbt
dbt debug
dbt run

08:47:27  Running with dbt=1.8.9
08:47:27  dbt version: 1.8.9
08:47:27  python version: 3.10.19
08:47:27  python path: /Users/spartan/venvs/lab2/bin/python3.10
08:47:27  os info: macOS-15.6.1-arm64-arm-64bit
08:47:28  Using profiles dir at /Users/spartan/Downloads/DATA226/LAB2new/dbt
08:47:28  Using profiles.yml file at /Users/spartan/Downloads/DATA226/LAB2new/dbt/profiles.yml
08:47:28  Using dbt_project.yml file at /Users/spartan/Downloads/DATA226/LAB2new/dbt/dbt_project.yml
08:47:28  adapter type: snowflake
08:47:28  adapter version: 1.8.4
08:47:28  Configuration:
08:47:28    profiles.yml file [OK found and valid]
08:47:28    dbt_project.yml file [OK found and valid]
08:47:28  Required dependencies:
08:47:28   - git [OK found]

08:47:28  Connection:
08:47:28    account: SFEDU02-LVB17920
08:47:28    user: COYOTE
08:47:28    database: USER_DB_COYOTE
08:47:28    warehouse: COYOTE_QUERY_WH
08:47:28    role: TRAINING_ROLE
08:47:28    schema: ANALYTICS
08:47:28    authenticator: None
08:47:28    oauth_client_id: None
08:47:28    query_tag: dbt-lab2
08:47:28    client_session_keep_alive: True
08:47:28    host: None
08:47:28    port: None
08:47:28    proxy_host: None
08:47:28    proxy_port: None
08:47:28    protocol: None
08:47:28    connect_retries: 1
08:47:28    connect_timeout: None
08:47:28    retry_on_database_errors: False
08:47:28    retry_all: False
08:47:28    insecure_mode: False
08:47:28    reuse_connections: None
08:47:28  Registered adapter: snowflake=1.8.4
08:47:31    Connection test: [OK connection ok]

08:47:31  All checks passed!
08:47:33  Running with dbt=1.8.9
08:47:33  Registered adapter: snowflake=1.8.4
08:47:34  Found 3 models, 2 data tests, 2 sources, 464 macros
08:47:34
08:47:37  Concurrency: 4 threads (target='dev')
08:47:37
08:47:37  1 of 3 START sql view model ANALYTICS_RAW.stg_stock_prices ..................... [RUN]
08:47:39  1 of 3 OK created sql view model ANALYTICS_RAW.stg_stock_prices ................ [SUCCESS 1 in 1.15s]
08:47:39  2 of 3 START sql view model ANALYTICS_ANALYTICS.int_with_indicators ............ [RUN]
08:47:40  2 of 3 OK created sql view model ANALYTICS_ANALYTICS.int_with_indicators ....... [SUCCESS 1 in 0.89s]
08:47:40  3 of 3 START sql table model ANALYTICS_ANALYTICS.fct_stock_analytics ........... [RUN]
08:47:42  3 of 3 OK created sql table model ANALYTICS_ANALYTICS.fct_stock_analytics ...... [SUCCESS 1 in 2.43s]
08:47:42
08:47:42  Finished running 2 view models, 1 table model in 0 hours 0 minutes and 8.19 seconds (8.19s).
08:47:42
08:47:42  Completed successfully
08:47:42
08:47:42  Done. PASS=3 WARN=0 ERROR=0 SKIP=0 TOTAL=3
```

VI. METRICS & CALCULATIONS

- Moving Averages: MA20 and MA50 calculated via window functions on CLOSE.
- RSI14: Implemented using gains/losses with 14-period smoothing; surfaced as RSI14.
- Daily Return: DAILY_RETURN = (CLOSE - LAG(CLOSE)) / LAG(CLOSE).
- Latest Close KPI: Latest CLOSE for a selected symbol (AAPL) via MAX(DATE) filter.

VII. SUPEREST DASHBOARDS

Stock Analytics Dashboard — Description

Purpose

This dashboard tracks daily U.S. equity performance for selected tickers and highlights short-term momentum vs. medium-term trend. It's designed to help you quickly answer: *Is the stock trending up or down? Is momentum stretched (overbought/oversold)? How volatile were recent sessions?*

Intended users

- Students and instructors reviewing Lab 2 outcomes
- Anyone needing a quick visual on price trend, momentum (RSI), and day-to-day returns

Datasets (Superset "Table / Dataset")

- **Primary table:**
  USER_DB_COYOTE.ANALYTICS_ANALYTICS.FCT_STOCK_ANALYTICS
  - **Columns typically used:**
    - DATE – trading date
    - SYMBOL – ticker (e.g., AAPL, MSFT, GOOG)
    - CLOSE – daily close price
    - MA20, MA50 – 20/50-day simple moving averages
    - RSI14 – 14-day Relative Strength Index
    - DAILY_RETURN – % change from prior close
- **Lineage:**
  Raw prices are ingested with Airflow → stored in Snowflake RAW/PUBLIC → modeled with dbt into ANALYTICS_RAW.STG_STOCK_PRICES → enriched into ANALYTICS_ANALYTICS.FCT_STOCK_ANALYTICS.

Refresh cadence

- **Airflow DAG #1** pulls prices daily and loads Snowflake.
- **dbt (DAG #2)** builds models and indicators after load.
- Refresh the Superset dataset (or set a cache TTL) to see the latest.

Dashboard tiles

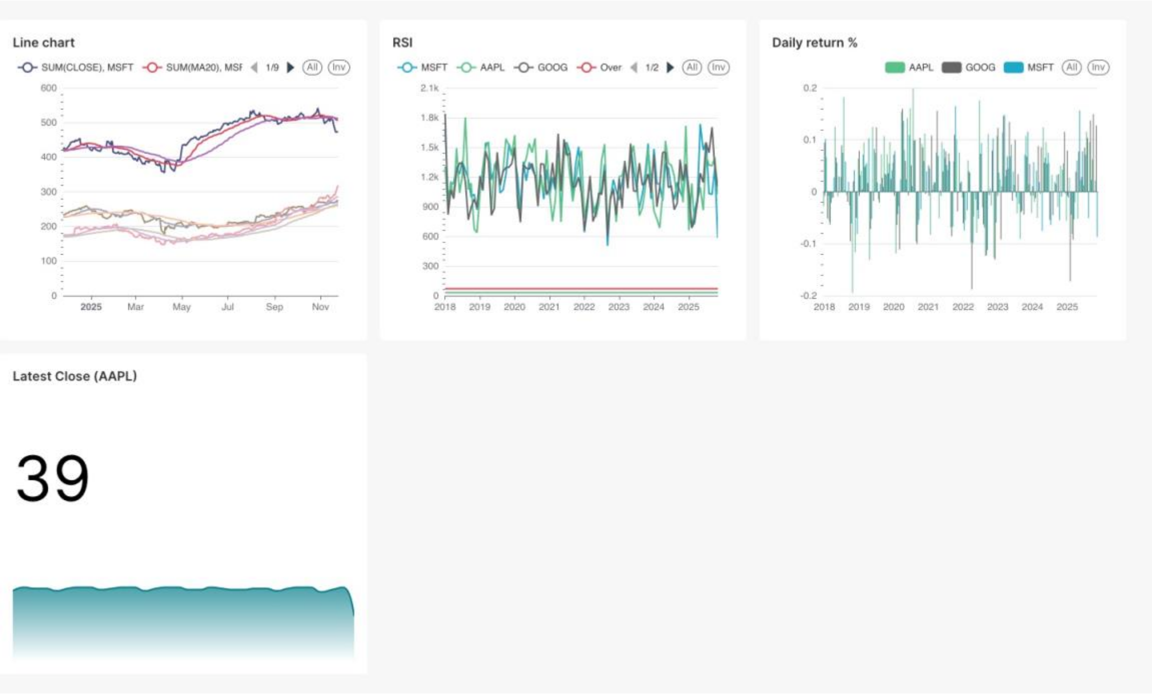
1. **Close Price vs. MA20/MA50 (Line)**
   - **What it shows:** Price trend with short- and medium-term moving averages.
   - **How to read:**
     - Price above both MAs → bullish momentum.
     - MA20 crossing above MA50 → potential bullish crossover.
2. **RSI (Line)**
   - **What it shows:** 14-day momentum oscillator.
   - **Guides:** Reference lines at **70** (overbought) and **30** (oversold).

    - **How to read:** Sustained >70 may precede pullbacks; <30 can indicate exhaustion.

3. **Daily Return % (Bar)**
    - **What it shows:** Session-to-session volatility and drift.
    - **How to read:** Clusters of large positive/negative bars indicate event-driven periods.
4. **Latest Close (KPI / Big Number)** *(example shown for AAPL)*
    - **What it shows:** Most recent close; optionally shows day-over-day delta.

We created the following visualizations from the fct_stock_analytics dataset:

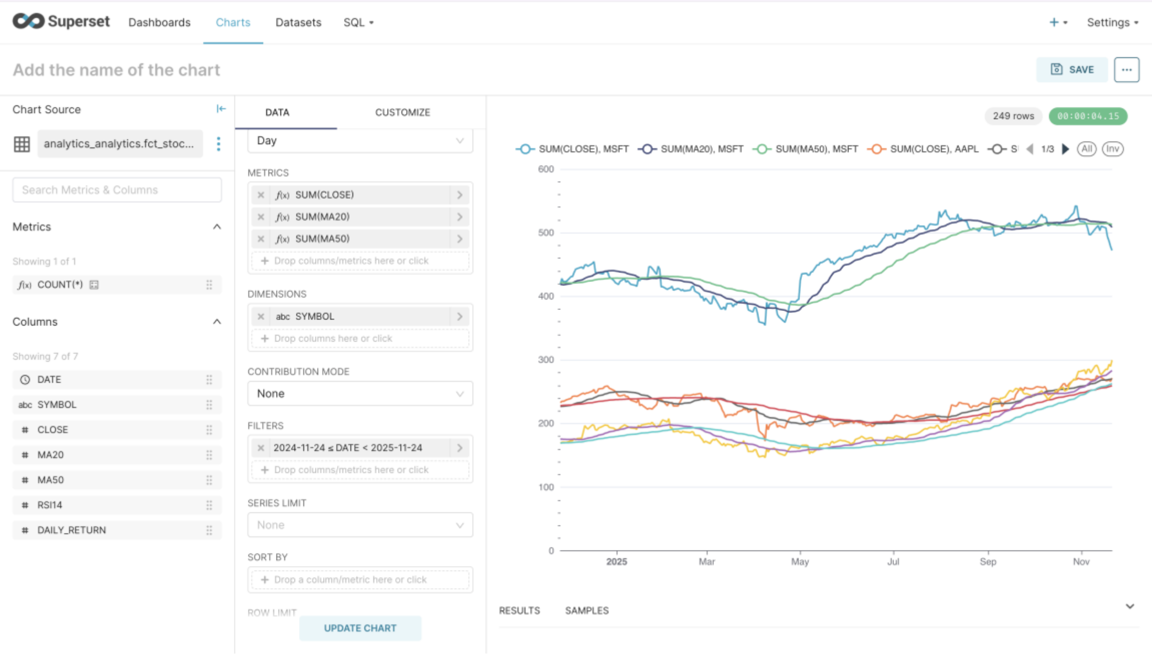
1) The Dashboard



Latest Close (AAPL)

39

2) Line Chart: CLOSE with MA20/MA50 per symbol over time.



Add the name of the chart

Chart Source: analytics_analytics.fct_stoc...

Metrics (Showing 1 of 1): COUNT(*)

Columns (Showing 7 of 7): DATE, SYMBOL, CLOSE, MA20, MA50, RSI14, DAILY_RETURN

DATA

Day

METRICS: SUM(CLOSE), SUM(MA20), SUM(MA50)

DIMENSIONS: SYMBOL

CONTRIBUTION MODE: None

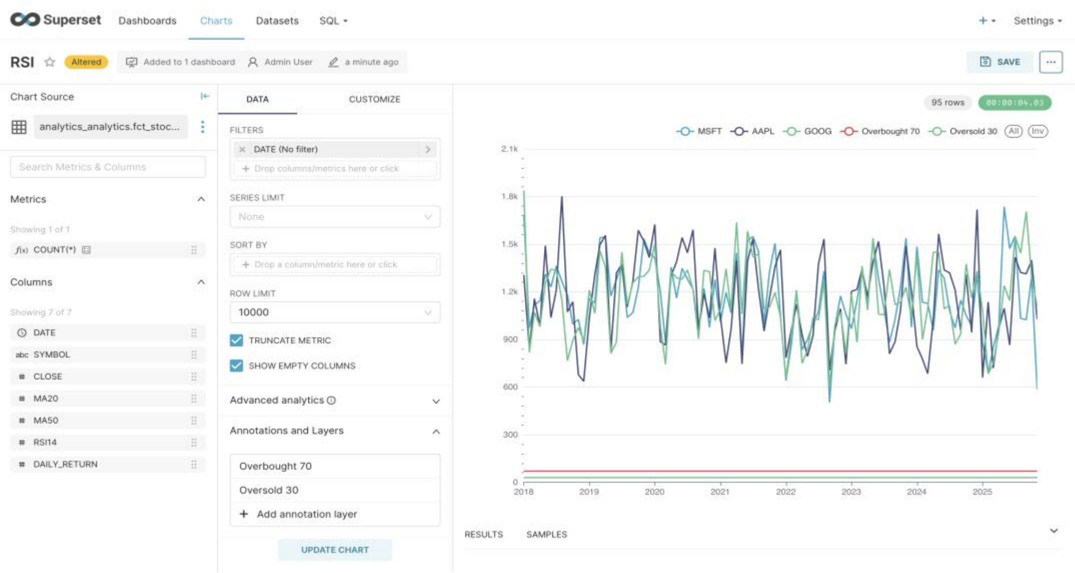
FILTERS: 2024-11-24 ≤ DATE < 2025-11-24

SERIES LIMIT: None

SORT BY: Drop a column/metric here or click

249 rows 00:00:04.15

3) RSI (Line): RSI14 with horizontal reference lines at 70 (overbought) and 30 (oversold), plus optional 50.



RSI · Altered · Added to 1 dashboard · Admin User · a minute ago

Chart Source: analytics_analytics.fct_stoc...

Metrics (Showing 1 of 1): COUNT(*)

Columns (Showing 7 of 7): DATE, SYMBOL, CLOSE, MA20, MA50, RSI14, DAILY_RETURN

DATA

FILTERS: DATE (No filter)

SERIES LIMIT: None

SORT BY: Drop a column/metric here or click

ROW LIMIT: 10000

☑ TRUNCATE METRIC

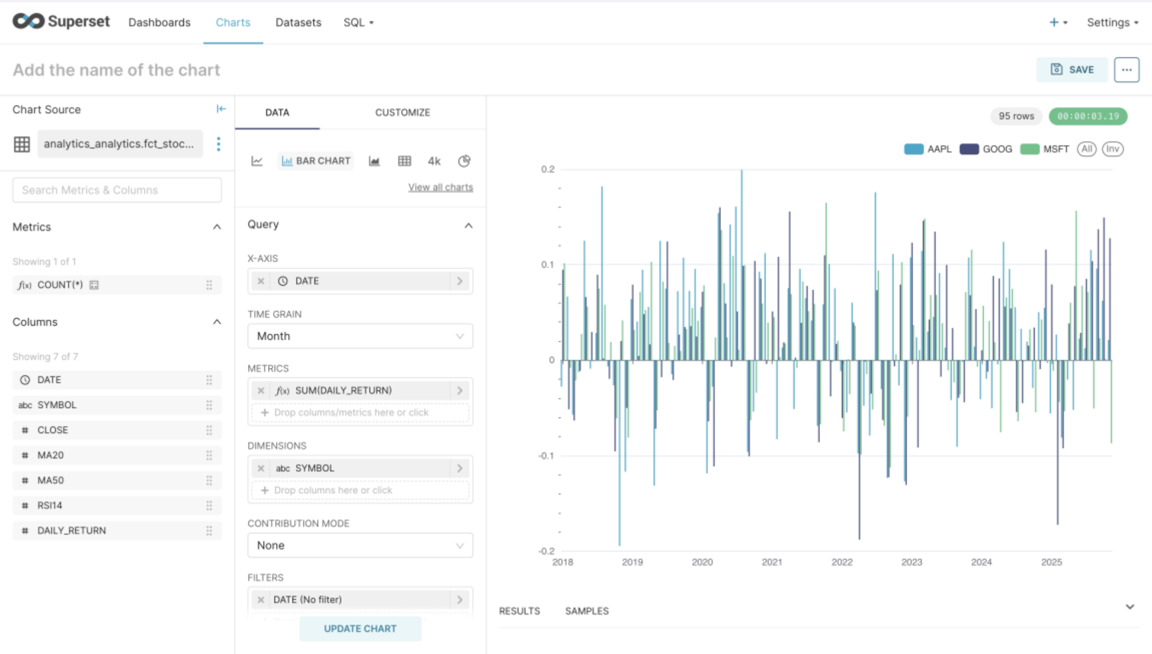
☑ SHOW EMPTY COLUMNS

Advanced analytics

Annotations and Layers: Overbought 70, Oversold 30

95 rows 00:00:04.03

4) Daily Return (%): Grouped bar chart by symbol, Y-axis formatted as percentage, zero reference line.



Add the name of the chart

Chart Source: analytics_analytics.fct_stoc...

Search Metrics & Columns

Metrics — Showing 1 of 1: COUNT(*)

Columns — Showing 7 of 7: DATE, SYMBOL, CLOSE, MA20, MA50, RSI14, DAILY_RETURN

Query

X-AXIS: DATE

TIME GRAIN: Month

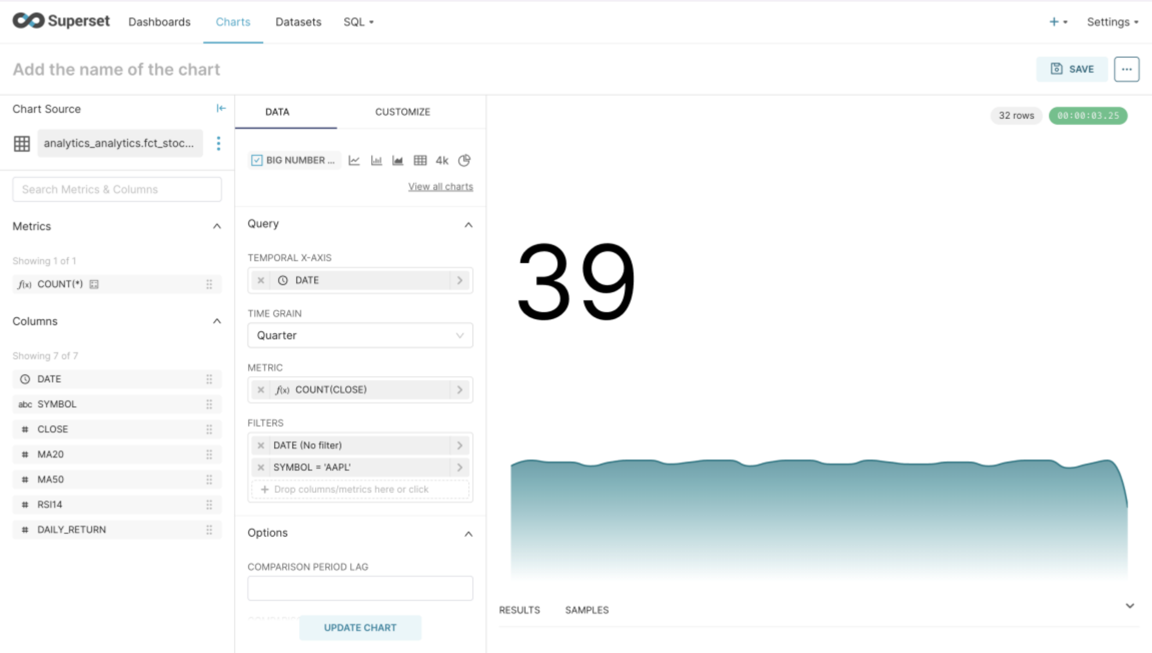
METRICS: SUM(DAILY_RETURN)

DIMENSIONS: SYMBOL

CONTRIBUTION MODE: None

FILTERS: DATE (No filter)

5) Latest Close (AAPL): Big Number showing the most recent CLOSE for AAPL. Screenshots are to be inserted in the placeholders below.



Add the name of the chart

Chart Source: analytics_analytics.fct_stoc...

Search Metrics & Columns

Metrics — Showing 1 of 1: COUNT(*)

Columns — Showing 7 of 7: DATE, SYMBOL, CLOSE, MA20, MA50, RSI14, DAILY_RETURN

Query

TEMPORAL X-AXIS: DATE

TIME GRAIN: Quarter

METRIC: COUNT(CLOSE)

FILTERS: DATE (No filter); SYMBOL = 'AAPL'

Options

COMPARISON PERIOD LAG

VIII. RESULTS & VALIDATION

Row counts and sanity queries verified the presence of 5,958 rows in STG_STOCK_PRICES and coherent outputs in the final fact table. The pipeline is idempotent with dbt materializations (views/tables) and is orchestrated via Airflow to ensure repeatability. KPI values and chart behavior were cross-checked for multiple symbols.

IX. DISCUSSION & LESSONS LEARNED

Key lessons include proper environment pinning for Superset dependencies, careful role naming (TRAINING_ROLE), and avoiding hardcoded credentials by using environment variables and Airflow Connections. dbt's layered approach helped maintain clarity between staging, intermediate logic, and final marts. Superset's annotation layers enabled domain-specific RSI thresholds.

X. FUTURE WORK

Potential extensions include: intraday granularity, adding additional technical indicators (MACD, Bollinger Bands), automated data freshness checks in Airflow, unit tests for SQL models, and parameterized dashboards for ad-hoc analysis.

APPENDIX A: REPRODUCIBILITY

1) Install & Configure dbt (example):

```
python3 -m venv ~/venvs/lab2
source ~/venvs/lab2/bin/activate
pip install dbt-snowflake==1.8.4
export DBT_PROFILES_DIR=/Users/spartan/Downloads/DATA226/LAB2new/dbt
dbt debug
```

2) profiles.yml (uses env vars):

```
lab2_snowflake:
  target: dev
  outputs:
    dev:
      type: snowflake
      account: "{{ env_var('SNOWFLAKE_ACCOUNT') }}"
      user: "{{ env_var('SNOWFLAKE_USER') }}"
      password: "{{ env_var('SNOWFLAKE_PASSWORD') }}"
      role: "{{ env_var('SNOWFLAKE_ROLE','TRAINING_ROLE') }}"
      warehouse: "{{ env_var('SNOWFLAKE_WAREHOUSE') }}"
      database: "{{ env_var('SNOWFLAKE_DATABASE') }}"
      schema: "{{ env_var('SNOWFLAKE_SCHEMA','ANALYTICS') }}"
      threads: 4
```

3) dbt_project.yml (schemas & materializations):

```
name: "lab2_stocks"
version: "1.0"
profile: "lab2_snowflake"
model-paths: ["models"]
seed-paths: ["seeds"]
target-path: "target"
clean-targets: ["target", "dbt_packages"]
models:
  lab2_stocks:
    +materialized: view
    staging:
      +schema: RAW
      +materialized: view
    intermediate:
      +schema: ANALYTICS
      +materialized: view
    marts:
      +schema: ANALYTICS
      +materialized: table
```

4) Superset (local) – minimal steps:

```
python3 -m venv ~/venvs/superset310
source ~/venvs/superset310/bin/activate
pip install "apache-superset==3.0.2" "sqlalchemy<2.0,>=1.4.49" "snowflake-sqlalchemy==1.7.7"
"numpy==1.23.5" "pandas<2.1,>=1.5.3" "backports.zstd<2"
export FLASK_APP="superset.app:create_app"
superset db upgrade
superset fab create-admin --username admin --firstname Admin --lastname User --email
admin@example.com --password admin
superset init
superset run -p 8088
```

REFERENCES

[1] dbt Labs. dbt Core & dbt-snowflake docs.

[2] Apache Airflow Documentation.

[3] Snowflake Documentation.

[4] Apache Superset Documentation.