

Name: Mrunali Katta

ID : 017516785

DATA 226 - Homework 05

Github link: <https://github.com/mrunalikatta1998/Data-Warehouse-and-Pipelines/tree/main/HW%205>

Created tasks using @task decorator

```
from airflow.decorators import dag, task
```

```
from airflow.providers.snowflake.hooks.snowflake import SnowflakeHook
```

```
from datetime import datetime, timedelta
```

```
import requests
```

```
import pandas as pd
```

```
# Default DAG arguments
```

```
default_args = {
```

```
    "owner": "airflow",
```

```
    "depends_on_past": False,
```

```
    "start_date": datetime(2024, 3, 1),
```

```
    "retries": 1,
```

```
    "retry_delay": timedelta(minutes=5),
```

```
}
```

```
@dag(
```

```
    dag_id="stock_price_to_snowflake",
```

```
    default_args=default_args,
```

```
    schedule_interval=None,
```

```
    catchup=False,
```

```
    tags=["snowflake"],
```

```
)
```

```
def stock_price_dag():
```

```
    @task
```

```
    def fetch_stock_data():
```

```
        """Fetch stock price data from Alpha Vantage API."""
```

```
        symbol = "AAPL"
```

```
        api_key = "VBB1QAY8METUTVJZ"
```

```
        url =
```

```
f"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={symbol}&apikey={api_key}&out
putsize=compact"
```

```
        response = requests.get(url)
```

```
        data = response.json()
```

```

time_series = data.get("Time Series (Daily)", {})
ninety_days_ago = (datetime.now() - timedelta(days=90)).strftime('%Y-%m-%d')

results = [
    {
        "symbol": symbol,
        "date": d,
        "open": time_series[d]["1. open"],
        "high": time_series[d]["2. high"],
        "low": time_series[d]["3. low"],
        "close": time_series[d]["4. close"],
        "volume": time_series[d]["5. volume"]
    }
    for d in time_series if d >= ninety_days_ago
]

return results

@task
def load_to_snowflake(data):
    """Load stock data into Snowflake, ensuring idempotency."""
    df = pd.DataFrame(data)

    # Secure Snowflake Connection
    snowflake_hook = SnowflakeHook(snowflake_conn_id="snowflake_conn")
    conn = snowflake_hook.get_conn()
    cursor = conn.cursor()

    # Ensure database and schema selection
    cursor.execute("USE DATABASE DEV;")
    cursor.execute("USE SCHEMA RAW;")

    # Create table if it doesn't exist
    create_table_sql = """
CREATE TABLE IF NOT EXISTS DEV.RAW.STOCK_PRICE (
    SYMBOL STRING,
    DATE DATE,
    OPEN FLOAT,
    HIGH FLOAT,
    LOW FLOAT,
    CLOSE FLOAT,
    VOLUME INT,
    PRIMARY KEY (SYMBOL, DATE)
);
"""

    cursor.execute(create_table_sql)

    try:
        cursor.execute("BEGIN;") # Start Transaction

```

```

# **Delete existing records for full refresh**
cursor.execute("DELETE FROM DEV.RAW.STOCK_PRICE;")

# Insert new records
insert_sql = """
INSERT INTO DEV.RAW.STOCK_PRICE (SYMBOL, DATE, OPEN, HIGH, LOW, CLOSE, VOLUME)
VALUES (%(symbol)s, %(date)s, %(open)s, %(high)s, %(low)s, %(close)s, %(volume)s);
"""

cursor.executemany(insert_sql, df.to_dict("records"))

# **COMMIT to save changes**
cursor.execute("COMMIT;")
cursor.close()

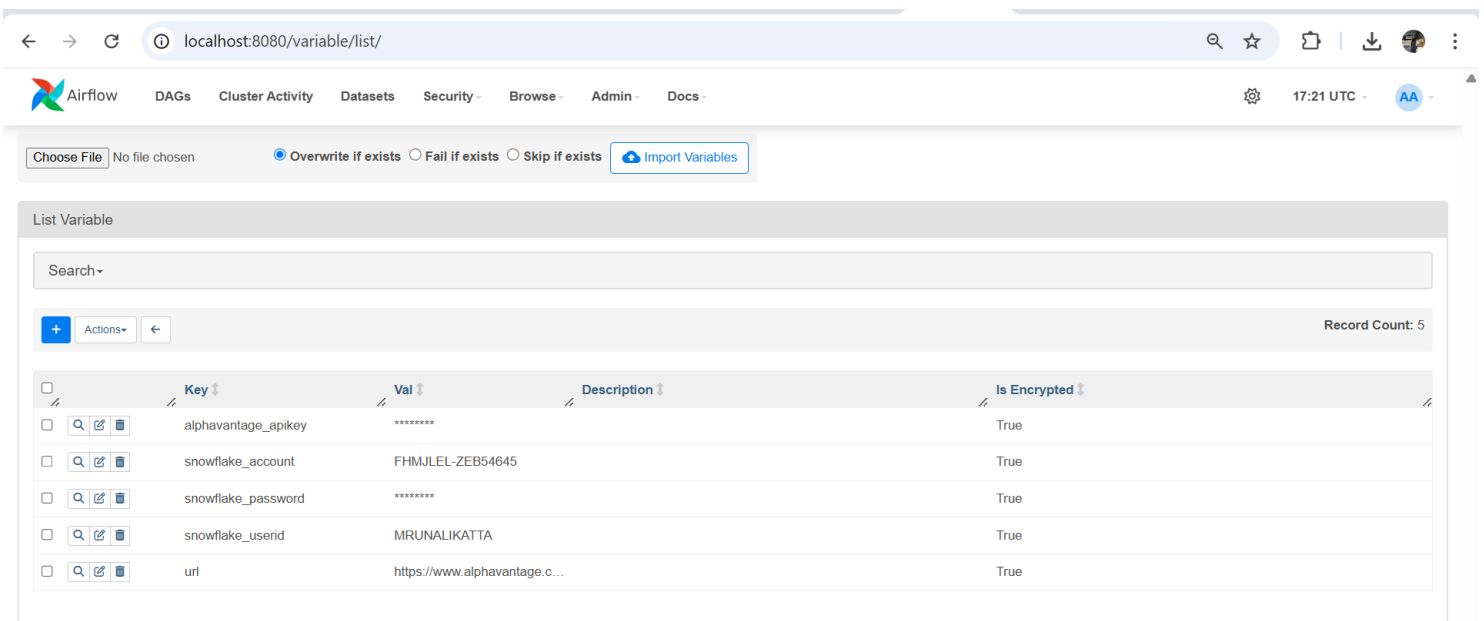
except Exception as e:
    cursor.execute("ROLLBACK;") # Undo changes on failure
    raise e

# DAG Task dependencies
stock_data = fetch_stock_data()
load_to_snowflake(stock_data)

stock_price_dag()

```


Set up a variable for Alpha Vantage API key



The screenshot shows the Apache Airflow web interface at localhost:8080/variable/list/. The page title is "List Variable". Below the title is a search bar and a "Record Count: 5" indicator. The main content is a table with the following columns: Key, Val, Description, and Is Encrypted. The table contains five rows of variables.

Key	Val	Description	Is Encrypted
alphavantage_apikey	*****		True
snowflake_account	FHMJLEL-ZEB54645		True
snowflake_password	*****		True
snowflake_userid	MRUNALIKATTA		True
url	https://www.alphavantage.c...		True

Set up Snowflake Connection

AIRFLOW

DAGS

Cluster Activity

Datasets

Security

Browse

Admin

Docs

Edit Connection

Connection Id *	snowflake_conn
Connection Type *	Snowflake
Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.	
Description	
Schema	RAW
Login	MRUNALIKATTA
Password	snowflake password
Extra	<pre>{ "account": "FHMJLEL-ZEB54645", "warehouse": "COMPUTE_WH", "database": "DEV", "role": "ACCOUNTADMIN", "insecure_mode": false }</pre>
Account	FHMJLEL-ZEB54645
Warehouse	COMPUTE_WH

☒ Auto-refresh Showing 1-2 of 2

Clear task Mark state as... Filter Tasks

(by attempts)

All Levels All File Sources ☐ Wrap [Download](#) [See More](#)

```
[2025-03-06, 09:24:13 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:13 UTC] {cursor.py:806} INFO - query: [INSERT INTO dev.raw.stock_price (symbol, date, open, high, low, close, volume) V...]
[2025-03-06, 09:24:13 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:13 UTC] {cursor.py:806} INFO - query: [INSERT INTO dev.raw.stock_price (symbol, date, open, high, low, close, volume) V...]
[2025-03-06, 09:24:14 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:14 UTC] {cursor.py:806} INFO - query: [INSERT INTO dev.raw.stock_price (symbol, date, open, high, low, close, volume) V...]
[2025-03-06, 09:24:14 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:14 UTC] {cursor.py:806} INFO - query: [INSERT INTO dev.raw.stock_price (symbol, date, open, high, low, close, volume) V...]
[2025-03-06, 09:24:15 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:15 UTC] {cursor.py:806} INFO - query: [INSERT INTO dev.raw.stock_price (symbol, date, open, high, low, close, volume) V...]
[2025-03-06, 09:24:15 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:15 UTC] {cursor.py:806} INFO - query: [INSERT INTO dev.raw.stock_price (symbol, date, open, high, low, close, volume) V...]
[2025-03-06, 09:24:16 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:16 UTC] {cursor.py:806} INFO - query: [INSERT INTO dev.raw.stock_price (symbol, date, open, high, low, close, volume) V...]
[2025-03-06, 09:24:16 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:16 UTC] {cursor.py:806} INFO - query: [INSERT INTO dev.raw.stock_price (symbol, date, open, high, low, close, volume) V...]
[2025-03-06, 09:24:16 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:16 UTC] {cursor.py:806} INFO - query: [INSERT INTO dev.raw.stock_price (symbol, date, open, high, low, close, volume) V...]
[2025-03-06, 09:24:17 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:17 UTC] {cursor.py:806} INFO - query: [COMMIT;]
[2025-03-06, 09:24:25 UTC] {cursor.py:819} INFO - query execution done
[2025-03-06, 09:24:25 UTC] {cursor.py:962} INFO - Number of results in first chunk: 1
[2025-03-06, 09:24:25 UTC] {logging_mixin.py:150} INFO - 100 records inserted successfully.
[2025-03-06, 09:24:25 UTC] {python.py:183} INFO - Done. Returned value was: None
[2025-03-06, 09:24:25 UTC] {taskinstance.py:1350} INFO - Marking task as SUCCESS, dag_id=stock_price_to_snowflake, task_id=load_to_snowflake, execution_d
[2025-03-06, 09:24:26 UTC] {local_task_job_runner.py:225} INFO - Task exited with return code 0
[2025-03-06, 09:24:26 UTC] {taskinstance.py:2653} INFO - 0 downstream tasks scheduled from follow-on schedule check
```