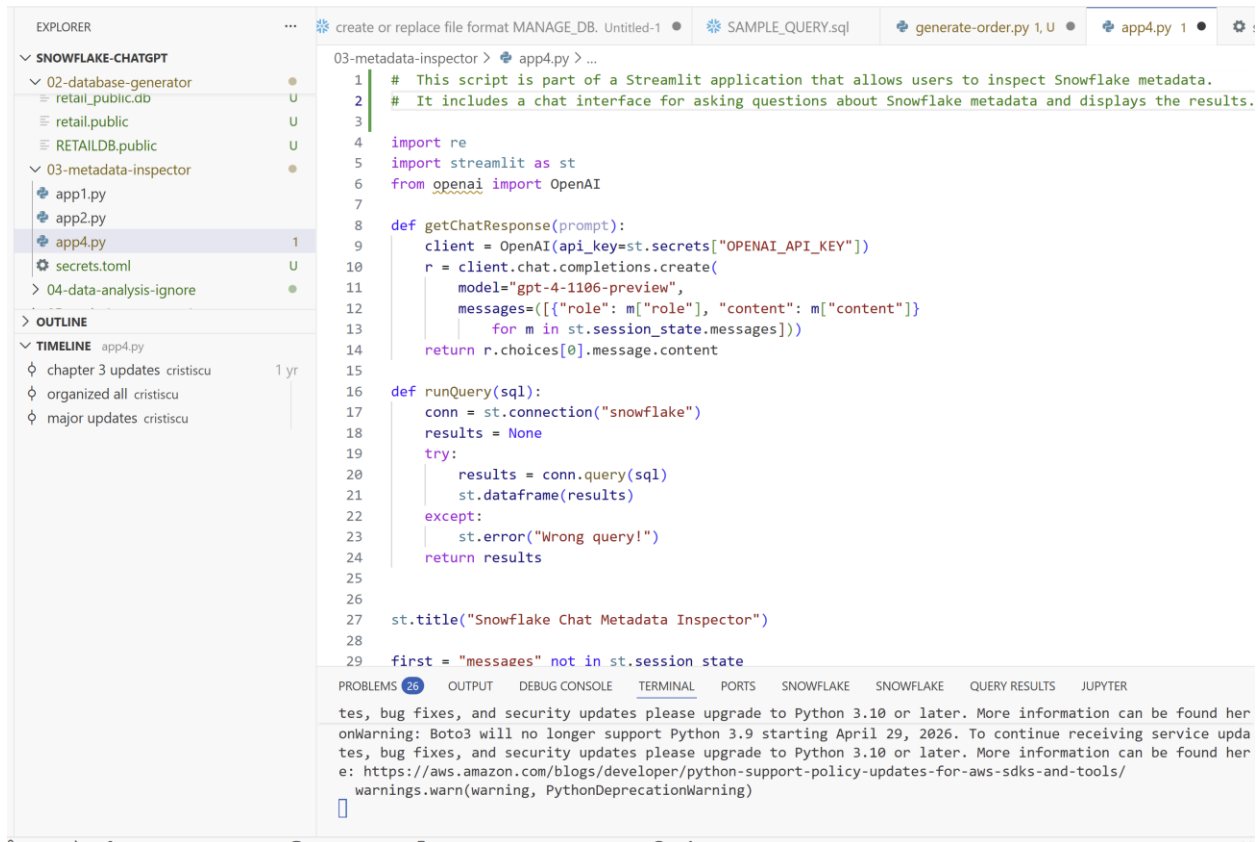


## Snowflake – Metadata Validation Script

This code turns Snowflake metadata into a conversational interface, allowing users to ask natural-language questions and instantly retrieve governance and catalog insights using live SQL. It helps teams understand data estates faster from table ownership and usage to schema structure without writing complex INFORMATION\_SCHEMA queries manually. In real projects, this is powerful for data discovery, impact analysis, audits, onboarding new engineers, and reducing dependency on platform experts while keeping everything governed and query-backed.



The screenshot displays a Jupyter Notebook environment with the following components:

- EXPLORER:** A sidebar on the left showing a file tree. It includes folders like '02-database-generator' and '03-metadata-inspector', and files such as 'app1.py', 'app2.py', 'app4.py', and 'secrets.toml'. The '03-metadata-inspector' folder is expanded, showing 'app4.py' selected.
- Code Editor:** The main area shows a Python script for a Streamlit application. The script is titled '03-metadata-inspector > app4.py > ...'. It includes comments, imports for 're', 'streamlit as st', and 'OpenAI', and defines two functions: 'getChatResponse(prompt)' and 'runQuery(sql)'. The 'getChatResponse' function uses the OpenAI API to generate responses based on a prompt. The 'runQuery' function connects to a Snowflake database and executes a query, returning the results as a DataFrame. The script also sets the title of the Streamlit application to 'Snowflake Chat Metadata Inspector' and initializes the session state.
- TERMINAL:** The bottom panel shows the output of the script. It displays a warning message from Boto3 about Python 3.9 support, followed by a message indicating that the script is part of a Streamlit application.

```
1 # This script is part of a Streamlit application that allows users to inspect Snowflake metadata.
2 # It includes a chat interface for asking questions about Snowflake metadata and displays the results.
3
4 import re
5 import streamlit as st
6 from openai import OpenAI
7
8 def getChatResponse(prompt):
9     client = OpenAI(api_key=st.secrets["OPENAI_API_KEY"])
10    r = client.chat.completions.create(
11        model="gpt-4-1106-preview",
12        messages=[{"role": "user", "content": m["content"]}
13                  for m in st.session_state.messages])
14    return r.choices[0].message.content
15
16 def runQuery(sql):
17     conn = st.connection("snowflake")
18     results = None
19     try:
20         results = conn.query(sql)
21         st.dataframe(results)
22     except:
23         st.error("Wrong query!")
24     return results
25
26
27 st.title("Snowflake Chat Metadata Inspector")
28
29 first = "messages" not in st.session_state
```


tes, bug fixes, and security updates please upgrade to Python 3.10 or later. More information can be found here  
onWarning: Boto3 will no longer support Python 3.9 starting April 29, 2026. To continue receiving service updates, bug fixes, and security updates please upgrade to Python 3.10 or later. More information can be found here: <https://aws.amazon.com/blogs/developer/python-support-policy-updates-for-aws-sdks-and-tools/>  
warnings.warn(warning, PythonDeprecationWarning)

←→

http://localhost:8501

Getting Started Dell McAfee Security Introducing Lakebridg... (3) Achilles Intro HD - ... Request #295484: Fw... Gmail YouTube Maps Resource classes for w...

Update We've introduced a Firefox [Terms of Use](#) and updated our [Privacy Notice](#). Please take a moment to review and accept. [Learn more](#) [Accept](#)



To retrieve a list of tables that exist in the `TPCH_SF1` schema within the `SNOWFLAKE_SAMPLE_DATA` database, you would issue the following query:

```
SELECT TABLE_NAME
FROM SNOWFLAKE_SAMPLE_DATA.INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'TPCH_SF1';
```

This query will provide you with the names of all tables within the `TPCH_SF1` schema in the `SNOWFLAKE_SAMPLE_DATA` database.

	TABLE_NAME
0	LINEITEM
1	NATION
2	CUSTOMER
3	REGION
4	SUPPLIER
5	PARTSUPP
6	PART
7	ORDERS

What tables exist in the `TPCH_SF1` schema?

▶