Building a Spy Agency Data Scavenger Hunt with Snowflake Cortex

Overview

Duration: 5

This quickstart will guide you through building an engaging spy agency-themed data scavenger hunt application using Snowflake Cortex. You'll create an interactive chat interface that allows users to query mission data, search for clues, and solve challenges using Snowflake's AI capabilities.

What You Will Build

A full-featured spy agency intelligence application that enables agents to:

- Query a database of spy mission reports and intercepted messages
- Use natural language to find specific intelligence data
- Solve intelligence challenges through interactive queries

What You Will Learn

- How to set up Snowflake Cortex Search for semantic document retrieval
- How to build a conversational AI interface with Streamlit in Snowflake
- How to use Claude models within Snowflake for intelligent responses

Prerequisites

- Snowflake account with access to a supported region for Cortex functions
- Account must have these features enabled:
  - Cortex LLM Functions
  - Streamlit in Snowflake
  - Cortex Search

Setting Up Your Environment

Duration: 10

Download Setup Files

First, download these required files from GitHub:

1. `setup.sql` - Contains all the SQL statements to create the database, tables, and sample data
2. `spy_report_semantic_model.yaml` - Contains the semantic model definition
3. `streamlit.py` - Contains the Streamlit application code

Run the Setup Script

1. Open a new SQL worksheet in Snowflake
2. Copy and paste the contents of `setup.sql` into the worksheet
3. Run the entire script

This script will:

- Create the spy_agency database and intel schema
- Create a warehouse called spy_agency_wh
- Create a stage for storing files
- Create tables for spy_missions and spy_reports
- Insert sample data into both tables
- Create a Cortex Search service called spy_mission_search

Upload the Semantic Model

1. Go to Data → Databases → SPY_AGENCY → INTEL → STAGE
2. Click "Upload" and select your `spy_report_semantic_model.yaml` file
3. Verify the upload with:

```
Unset
LIST @spy_agency.intel.stage;
```

Creating the Streamlit Application

Duration: 8

Setting Up the Streamlit App

To create and configure your Streamlit application in Snowflake:

1. Navigate to Streamlit in Snowflake:
    - Click on the **Streamlit** tab in the left navigation pane
    - Click on **+ Streamlit App** button in the top right
2. Configure App Settings:

- Enter "Spy Agency Intelligence Portal" as your app name
- Select spy_agency_wh as your warehouse
- Choose spy_agency as your database and intel as your schema

3. Create the app:

    - In the editor, paste the complete code from the `streamlit.py` file
    - Click "Run" to launch your application

Challenge 1: A Suspicious Message

Duration: 5

Objective

Find an intercepted message about a security breach.

Challenge Setup

This challenge tests users' ability to search messages for specific content. The user needs to find evidence of a possible security breach in the intercepted messages.

Hint 1

"One of our agents believes there's a mole in the network. Try searching intercepted messages for anything related to a breach or security warning."

Suggested search terms: "mole", "breach", "security threat"

Hint 2

"The agent was concerned about infiltration—look for messages where someone warns about a possible breach."

Suggested search terms: "possible breach", "suspect a mole"

Hint 3 (Final)

"Search for the exact phrase 'Possible breach'—this message will reveal who raised the alarm."

Solution

Intercepted message: "Possible breach. Suspect a mole in the network."

Location: M003 (Paris, sent by K9Q)

Testing the Challenge

Try typing this query into your Streamlit app: "Find any messages about a security breach"

Challenge 2: Identifying the Double Agent

Duration: 5

Objective

Query spy_reports for the agent flagged as a double agent in M003.

Challenge Setup

This challenge tests users' ability to query structured data. They need to identify which agent was flagged as a double agent in a specific mission.

Hint 1

"The Paris mission was flagged as compromised. Check mission reports to see if any agents were suspected."

Suggested SQL query: "Which agent was suspected in the Paris mission?"

Hint 2

"One agent was marked as a double agent in that mission. Try checking which agent was flagged."

Suggested SQL query: "Show the suspected double agent for Mission M003"

Hint 3 (Final)

"Check the report for Mission M003—Agent K9Q was marked as the suspected double agent."

Solution

Agent: K9Q is the suspected double agent.

Confirmed in: Mission reports for M003.

Testing the Challenge

Try typing this query into your Streamlit app: "Which agent was suspected as a double agent in mission M003?"

Challenge 3: Tracking the Rogue Spy

Duration: 5

Objective

First, check spy_reports for the most recent compromised mission, then search spy_missions for relevant intercepted messages.

Challenge Setup

This challenge tests users' ability to combine multiple queries. They need to identify a recent compromised mission and then find related messages.

Hint 1

"If a double agent is involved, other missions may have been compromised. Check reports for recent failed missions."

Suggested SQL query: "Show the most recent compromised mission"

Hint 2

"A recent mission involved heightened security measures. Look for a message mentioning security lockdowns."

Suggested search terms: "secure the perimeter", "lockdown", "restricted access"

Hint 3 (Final)

"Search for 'Secure the perimeter'—this intercepted message will tell you where the rogue spy was last seen."

Solution

Latest compromised mission: M016 (Istanbul, involving A1D).

Intercepted message: "Secure the perimeter. No one gets in or out."

Testing the Challenge

Try these queries in sequence:

1. "What was the most recent compromised mission?"
2. "Find messages about securing the perimeter"

Challenge 4: Decoding the Spy's Plan

Duration: 5

Objective

Find the rogue spy's final encrypted message to uncover their last move.

Challenge Setup

This is the final challenge that tests users' ability to find specific encrypted communications. They need to locate the spy's final message to discover their plan.

Hint 1

"The rogue spy must have left an encrypted message before disappearing. Search for transmissions that mention encryption or secret keyphrases."

Suggested search terms: "high encryption", "classified message", "keyphrase"

Hint 2

"A critical encrypted message contains the spy's final location. Look for words like vault, hidden package, or a secret phrase."

Suggested search terms: "vault", "package", "keyphrase"

Hint 3 (Final)

"Search for 'Keyphrase: Golden Dawn'—this will uncover the spy's final move in Athens."

Solution

Final encrypted message: "The package is in the vault. Keyphrase: Golden Dawn."

Location: M019 (Athens, A1D's final encrypted message).

Testing the Challenge

Try typing this query into your Streamlit app: "Find messages containing a keyphrase"

Conclusion and Resources

Duration: 2

Conclusion

Congratulations! You've successfully built a spy agency intelligence portal using Snowflake's Cortex capabilities and created an engaging data scavenger hunt. This application demonstrates how to combine data processing with AI features to create an interactive learning experience using natural language queries.

What You Learned

- How to set up a Snowflake environment for AI-powered applications
- How to use Cortex Search for semantic document retrieval
- How to build a Streamlit interface that leverages Claude models within Snowflake
- How to create an interactive challenge-based learning experience
- How to structure data and queries to support narrative-driven exploration

Resources

Getting Started with Cortex Agents:

- Getting Started with Cortex Agents
- Getting Started with Snowflake Cortex Agents API and React
- Getting Started with Cortex Agents and Slack
- Getting Started with Cortex Agents and Microsoft Teams