ExamsIndex
Practice Materials & Tests

DEMO VERSION

Snowflake

GES-C01 Exam

SnowPro® Specialty: Gen AI Certification Exam

Exam Latest Version: 6.1

Question 1. (Multi Select)

A Snowflake developer, AI_ENGINEER, is creating a Streamlit in Snowflake (SiS) application that will utilize a range of Snowflake Cortex LLM functions, including SNOWFLAKE.CORTEX.COMPLETE, SNOWFLAKE.CORTEX.CLASSIFY_TEXT, and SNOWFLAKE.CORTEX.EMBED_TEXT_768. The application also needs to access data from tables within a specific database and schema. AI_ENGINEER has created a custom role, app_dev_role, for the application to operate under. Which of the following privileges or roles are absolutely necessary to grant to app_dev_role for the successful execution of these Cortex LLM functions and interaction with the specified database objects? (Select all that apply.)

- ☐ The SNOWFLAKE.CORTEX_USER database role, which provides the necessary permissions to call Snowflake Cortex AI functions.
- ☐ The CREATE SNOWFLAKE.ML.DOCUMENT_INTELLIGENCE privilege on the schema where the application resides.
- ☐ The USAGE privilege on the specific database and schema where the Streamlit application and its underlying data tables are located.
- ☐ The ACCOUNTADMIN role to ensure unrestricted access to all Snowflake Cortex features.
- ☐ The CREATE COMPUTE POOL privilege to provision resources for the Streamlit application.

A: Option A
B: Option B
C: Option C
D: Option D
E: Option E

Correct Answer: A, C

Explanation:

To execute Snowflake cortex AI functions such as 'SNOWFLAKE.CORTEX.COMPLETE , 'SNOWFLAKE.CORTEX.CLASSIFY_TEXT, and 'EMBED_TEXT_768' (or their SAE prefixed counterparts), the role used by the application in this case) must be granted the 'SNOWFLAKE.CORTEX_USER database role. Additionally, for the Streamlit application to access any database or schema objects (like tables for data input/output, or for the Streamlit app itself if it is stored as a database object), the USAGE privilege must be granted on those specific database and schema objects. Option B, 'CREATE SNOWFLAKE.ML.DOCUMENT_INTELLIGENCE, is a privilege specific to creating Document AI model builds and is not required for general Cortex LLM functions. Option D, 'ACCOUNTADMIN', grants excessive privileges and is not a best practice for application roles. Option E, 'CREATE COMPUTE POOL' , is a privilege related to Snowpark Container Services for creating compute pools, which is not directly required for running a Streamlit in Snowflake application that consumes Cortex LLM functions.

Question 2. (Single Select)

A data application developer is tasked with building a multi-turn conversational AI application using Streamlit in Snowflake (SiS) that leverages the COMPLETE (SNOWFLAKE. CORTEX) LLM function. To ensure the conversation flows naturally and the LLM maintains context from previous interactions, which of the following is the most appropriate method for handling and passing the conversation history?

- ○ The developer should store the entire conversation history in a temporary table in Snowflake and query it with each new turn, passing only the latest user message to the `COMPLETE` function.
- ○ Snowflake automatically manages conversational context for `COMPLETE` within the session, so the developer only needs to pass the current user prompt as a string.
- ○ The conversation history must be explicitly managed within the Streamlit application's state, typically by initializing `st.session_state.messages = []` and appending each user and assistant message as an object with `'role'` and `'content'` keys, then passing the full list to the `prompt_or_history` argument of `COMPLETE`.
- ○ The developer should concatenate all previous user prompts and assistant responses into a single, long string, and pass this as the `<prompt>` argument to `COMPLETE` for each turn.
- ○ The `COMPLETE` function has an optional `'conversation_id'` parameter that automatically retrieves and manages conversation history when provided.

A: Option A
B: Option B
C: Option C
D: Option D
E: Option E

Correct Answer: C

Question 3. (Multi Select)

A Snowflake developer, `AI_ENGINEER`, is creating a Streamlit in Snowflake (SiS) application that will utilize a range of Snowflake Cortex LLM functions, including `SNOWFLAKE.CORTEX.COMPLETE`, `SNOWFLAKE.CORTEX.CLASSIFY_TEXT`, and `SNOWFLAKE.CORTEX.EMBED_TEXT_768`. The application also needs to access data from tables within a specific database and schema. `AI_ENGINEER` has created a custom role, `app_dev_role`, for the application to operate under. Which of the following privileges or roles are absolutely necessary to grant to `app_dev_role` for the successful execution of these Cortex LLM functions and interaction with the specified database objects? (Select all that apply.)

A:

The `SNOWFLAKE.CORTEX_USER` database role, which provides the necessary permissions to call Snowflake Cortex AI functions.

B:

The CREATE SNOWFLAKE.ML.DOCUMENT_INTELLIGENCE privilege on the schema where the application resides.

C: The USAGE privilege on the specific database and schema where the Streamlit application and its underlying data tables are located.
D: The ACCOUNTADMIN role to ensure unrestricted access to all Snowflake Cortex features.
E: The CREATE COMPUTE POOL privilege to provision resources for the Streamlit application.

Correct Answer: A, C

The SNOWFLAKE.CORTEX_USER database role, which provides the necessary permissions to call Snowflake Cortex AI functions.

Explanation:

To execute Snowflake Cortex AI functions such as 'SNOWFLAKE.CORTEX.COMPLETE, 'CLASSIFY TEXT (SNOWFLAKE.CORTEX)' , and (or their prefixed counterparts like 'AI_COMPLETE' , 'AI_CLASSIFY , 'AI_EMBED), the role used by the application in this case) must be granted the database role. This role includes the privileges to call these functions. Additionally, for the Streamlit application to access any database or schema objects (like tables for data input/output, or for the Streamlit app itself if it is stored as a database object), the 'USAGE' privilege must be granted on those specific database and schema objects. Option B, 'CREATE SNOWFLAKE.ML.DOCUMENT_INTELLIGENCE , is a privilege specific to creating DocumentAI model builds and is not required for general Cortex LLM functions. Option D, 'ACCOUNTADMIN", grants excessive privileges and is not a best practice for application roles. Option E, 'CREATE COMPUTE POOL' , is a privilege related to Snowpark Container Services for creating compute pools, which is generally not directly required for running a Streamlit in Snowflake application that consumes Cortex LLM functions via SQL, unless the LLMs themselves were deployed as services on compute pools using Model Serving in Snowpark Container Services, which is not explicitly stated as the method of LLM usage here.

Question 4. (Single Select)

A data engineer is building a Snowflake data pipeline to ingest customer reviews from a raw staging table into a processed table. For each review, they need to determine the overall sentiment (positive, neutral, negative) and store this as a distinct column. The pipeline is implemented using SQL with streams and tasks to process new dat

a. Which Snowflake Cortex LLM function, when integrated into the SQL task, is best suited for this sentiment classification and ensures a structured, single-label output for each review?

- Use `SNOWFLAKE.CORTEX.SENTIMENT()` to get a numerical score, then apply a separate SQL CASE statement or UDF for classification into 'positive', 'neutral', 'negative'.
- Use `SNOWFLAKE.CORTEX.CLASSIFY_TEXT()` with the input text and a list of categories like `['positive', 'negative', 'neutral']` to directly obtain the classification label.
- Use `AI_COMPLETE()` with a prompt such as `'Classify the sentiment of this review: [review_text] into positive, neutral, or negative.'` and configure the `response_format` to be a string.
- Use `SNOWFLAKE.CORTEX.EXTRACT_ANSWER()` asking the question `'What is the sentiment of this review?'` to get a descriptive answer that then needs parsing.
- Use multiple `AI_FILTER()` calls, one for each sentiment category (e.g., `AI_FILTER(prompt('Is this review positive?'))`), and combine the boolean results.

A: Option A
B: Option B
C: Option C
D: Option D
E: Option E

Correct Answer: B

Explanation:

To classify text into predefined categories, the function (or its updated version, is purpose-built and directly returns the classification label. This approach is more direct and efficient than using 'SENTIMENT()' which returns a score, which extracts an answer to a question, or multiple calls which return Boolean values. While could be prompted for classification, is a more specific task-specific function designed for this exact use case within Cortex LLM functions.

Question 5. (Single Select)

A financial services company is developing an automated data pipeline in Snowflake to process Federal Reserve Meeting Minutes, which are initially loaded as PDF documents. The pipeline needs to extract specific entities like the FED's stance on interest rates ('hawkish', 'dovish', or 'neutral') and the reasoning behind it, storing these as structured JSON objects within a Snowflake table. The goal is to ensure the output is always a valid JSON object with predefined keys. Which AI_COMPLETE configuration, used within an in-line SQL statement in a task, is most effective for achieving this structured extraction directly in the pipeline?

- Using a simple prompt like `'Extract FED stance on interest rates and the reasoning as JSON from the document: [document_text]'` and expecting the LLM to format it correctly without further configuration.
- Setting the `temperature` parameter to 0 and `max_tokens` to a sufficiently large value, along with a prompt to output JSON, to inherently guide the LLM to a structured format.
- Employing the `response_format` argument within `AI_COMPLETE` with a JSON schema that explicitly defines the `stance` (enum: 'hawkish', 'dovish', 'neutral') and `reasoning` (string) fields, ensuring strict adherence to the output structure.
- Utilizing multiple calls to `SNOWFLAKE.CORTEX.EXTRACT_ANSWER()` or `AI_EXTRACT()`, one for the stance and another for the reasoning, then manually constructing the JSON in a subsequent SQL step.
- Relying on default `AI_COMPLETE` behavior, as Snowflake Cortex automatically detects the need for JSON output when entity extraction with specific fields is implied by the prompt.

A: Option A
B: Option B
C: Option C
D: Option D
E: Option E

Correct Answer: C

Explanation:

To ensure that LLM responses adhere to a predefined JSON structure, the 'AI_COMPLETE function's ‘response_format’ argument, which accepts a JSON schema, is the most effective and direct method. This mechanism enforces the structure, data types, and required fields, significantly reducing the need for post-processing and ensuring deterministic, high-quality output. The AI-Infused Data Pipelines with Snowflake Cortex blog highlights asking the LLM to create a JSON object for maximizing utility. While setting ‘temperature' to 0 can improve consistency, it does not enforce a specific schema. Prompt engineering (Option A) can help but does not guarantee strict adherence. Using multiple extraction calls (Option D) is less efficient and robust for extracting multiple related fields than a single 'AI_COMPLETE call with a structured output schema. Snowflake Cortex does not automatically infer and enforce a JSON schema without explicit configuration (Option E).

ExamsIndex

Demo PDF Complete

Your GES-C01 Demo (5 Questions)

Get the Complete Version

Full Questions with Detailed Explanations

Interactive Web-Based Exams Available

To get 50% off, use Coupon Code: OCT50

https://examsindex.com/exam/ges-c01