

# Brain Dump GES-C01 Free, Test GES-C01 Preparation



BONUS!!! Download part of ITPassLeader GES-C01 dumps for free: [https://drive.google.com/open?id=1-rvViD\\_XFN5qd9flSy6PefbT8NFbwRUM](https://drive.google.com/open?id=1-rvViD_XFN5qd9flSy6PefbT8NFbwRUM)

In order to adapt to different level differences in users, the GES-C01 exam questions at the time of writing teaching materials with a special focus on the text information expression, as little as possible the use of crude esoteric jargon, as much as possible by everyone can understand popular words to express some seem esoteric knowledge, so that more users through the GES-C01 Prep Guide to know that the main content of qualification examination, stimulate the learning enthusiasm of the user, arouse their interest in learning.

The high pass rate of our GES-C01 exam guide is not only a reflection of the quality of our learning materials, but also shows the professionalism and authority of our expert team on GES-C01 practice engine. Therefore, we have the absolute confidence to provide you with a guarantee: as long as you use our GES-C01 Learning Materials to review, you can certainly pass the exam, and if you do not pass the GES-C01 exam, we will provide you with a full refund.

>> **Brain Dump GES-C01 Free** <<

## Quiz Trustable GES-C01 - Brain Dump SnowPro® Specialty: Gen AI Certification Exam Free

Our product boosts many merits and functions. You can download and try out our GES-C01 test question freely before the purchase. You can use our product immediately after you buy our product. We provide 3 versions for you to choose and you only need 20-30 hours to learn our GES-C01 training materials and prepare the exam. The passing rate and the hit rate are both high. The purchase procedures are safe and we protect our client's privacy. We provide 24-hours online customer service and free update within one year. If you fail in the exam, we will refund you immediately. All in all, there are many advantages of our GES-C01 Training Materials.

## Snowflake SnowPro® Specialty: Gen AI Certification Exam Sample Questions (Q49-Q54):

### NEW QUESTION # 49

An operations team is investigating an issue with a generative AI application powered by Snowflake Cortex Analyst, where users reported unexpected behavior in generated SQL. To diagnose the problem, they examine the detailed event logs captured by Snowflake AI Observability. Which categories of information can they expect to find in these event tables to assist their investigation?

- A. The exact SQL queries that Cortex Analyst generated in response to user questions.
- B. The complete request and response bodies associated with the application's execution steps.
- C. The full text of the natural language questions submitted by the users.
- D. Any error messages or warnings that occurred during the processing of the request.
- E. Real-time CPU and memory usage statistics for the Snowflake virtual warehouse executing the LLM inference.

**Answer: A,B,C,D**

Explanation:

Cortex Analyst logs requests to an event table to aid in refining semantic models or views. These logs are comprehensive and include specific details crucial for debugging and monitoring. The captured information includes 'The user who asked the question', 'The question asked', 'Generated SQL', 'Errors and/or warnings', 'Request and response bodies', and 'Other metadata'. Therefore, options A, B, C, and D are all accurate descriptions of the data available in these event logs. Option E, real-time CPU and memory usage, refers to infrastructure monitoring metrics rather than the content specifically logged within the application's execution event table by Cortex Analyst itself.

#### NEW QUESTION # 50

A data engineering team is designing a Snowflake data pipeline to automatically enrich a 'customer issues' table with product names extracted from raw text-based 'issue\_description' columns. They want to use a Snowflake Cortex function for this extraction and integrate it into a stream and task-based pipeline. Given the 'customer\_issues' table with an 'issue\_id' and (VARCHAR), which of the following SQL snippets correctly demonstrates the use of a Snowflake Cortex function for this data enrichment within a task, assuming is a stream on the 'customer\_issues' table?

- A. Option D
- B. Option E
- C. Option C
- **D. Option B**
- E. Option A

**Answer: D**

Explanation:

Option B correctly uses to pull specific information (product name) from unstructured text, which is a common data enrichment task. It also integrates with a stream ('issue\_stream') by filtering for 'METADATA\$ACTION = 'INSERT'' and uses a 'MERGE' statement, which is suitable for incremental updates in a data pipeline by inserting new extracted data based on new records in the stream. Option A uses for generating a response, not for specific entity extraction, and its prompt is less precise for this task than 'EXTRACT\_ANSWER'. Option C uses 'SNOWFLAKE.CORTEX.CLASSIFY\_TEXT' for classification, not direct entity extraction of a product name, and attempts to update the source table directly, which is not ideal for adding new columns based on stream data. Option D proposes a stored procedure and task, which is a valid pipeline structure. However, the EXTRACT\_ANSWER call within the procedure only returns a result set and does not demonstrate the final insertion or merging step required to persist the extracted data into an 'enriched\_issues' table. Option E uses to generate vector embeddings, which is a form of data enrichment, but the scenario specifically asks for 'product names' (a string value), not embeddings for similarity search.

#### NEW QUESTION # 51

An ML engineer is preparing a Docker image for a custom LLM application that will be deployed to Snowpark Container Services (SPCS). The application uses a mix of packages, some commonly found in the Snowflake Anaconda channel and others from general open-source repositories like PyPI. They have the following Docker-file snippet and need to ensure the dependencies are correctly installed for the SPCS environment to support a GPU workload. Which of the following approaches for installing Python packages in the Dockerfile would ensure a robust and compatible setup for a custom LLM running in Snowpark Container Services, based on best practices for managing dependencies in this environment?

- **A.**
- B.
- C.
- D.
- E.

**Answer: A**

Explanation:

Option B is correct. The provided Dockerfile example for deploying Llama 2 in Snowpark Container Services explicitly uses 'conda install -n rapids -c https://repo.anaconda.com/pkg/snowflake' to install Snowflake-specific packages like 'snowflake-ml-python' and 'snowflake-snowpark-python' from the Snowflake Anaconda channel. It then uses 'pip install' for other open-source libraries that are not available or preferred from the Anaconda channels. Option A is incorrect because while pip can install many packages, the provided example demonstrates using 'conda' from the Snowflake Anaconda channel for certain foundational packages. Option C is incorrect because while 'conda-forge' is a common channel for open-source packages, the specific Snowflake-related packages in the example are pulled directly from the 'https://repo.anaconda.com/pkg/snowflake' channel. Although Source notes that 'conda-

forge' is assumed for 'conda\_dependencies' in when building container images, a Dockerfile explicitly defining 'RUN conda install' can specify the channel, which the example in demonstrates. Option D is incorrect because the 'defaultS channel often requires user acceptance of Anaconda terms, which is not feasible in an automated build environment. Option E is a generic approach for pip dependencies but doesn't specifically address the recommended use of 'conda' from the Snowflake Anaconda channel for certain core Snowflake packages as shown in the practical example.

#### NEW QUESTION # 52

Considering Snowflake's Gen AI principles for cost governance within Snowflake Cortex, an ML engineer is assessing the expenditure for an LLM fine-tuning job. Which option correctly identifies how compute costs for Cortex Fine-tuning are primarily incurred and how fine-tuned models are treated regarding usage by other customers?

- A. Costs are incurred per hour of compute pool usage, similar to virtual warehouses. Fine-tuned models are anonymized and used to train future foundation models for all customers.
- **B. Compute costs for fine-tuning are based on the number of tokens used in training, calculated as 'number of input tokens number of epochs trained'. Fine-tuned models built using a customer's data are available exclusively for that customer's use.**
- C. Fine-tuning costs are a flat monthly fee, irrespective of token usage or model size. Fine-tuned models become part of Snowflake's proprietary models after training.
- D. Costs are based on the number of fine-tuning jobs created, not tokens. Fine-tuned models are shared across all Snowflake customers to improve the general service.
- E. Only inference using fine-tuned models incurs costs, not the training itself. Fine-tuned models can be openly shared on the Snowflake Marketplace.

**Answer: B**

Explanation:

Snowflake Cortex Fine-tuning incurs compute cost based on the number of tokens used in training. Specifically, fine-tuning trained tokens are calculated as 'number of input tokens number of epochs trained'. Furthermore, fine-tuned models built using your data are available exclusively for your use and are not used to train, re-train, or fine-tune Models made available to others.

#### NEW QUESTION # 53

A development team is implementing a suite of generative AI applications on Snowflake, utilizing both SQL functions and the Cortex REST API. They prioritize content safety and plan to integrate Cortex Guard wherever possible. Considering the various interfaces for interacting with Snowflake Cortex LLMs, which of the following interfaces and functions support the direct use of Cortex Guard via the 'guardrails' argument or equivalent configuration?

- A. The 'SNOWFLAKE.CORTEX.CLASSIFY\_TEXT SQL function for text classification tasks.
- **B. The 'SNOWFLAKE.CORTEX.TRY\_COMPLETE SQL function, which is the error-tolerant version of 'COMPLETE.**
- **C. The 'Cortex Playground' (Public Preview) when testing prompts and model settings.**
- **D. The 'SNOWFLAKE.CORTEX.COMPLETE SQL function for generative AI tasks.**
- **E. The Snowflake Cortex LLM REST API when invoking the '/api/v2/cortex/inference:complete' endpoint.**

**Answer: B,C,D,E**

Explanation:

Cortex Guard is a feature specifically designed to filter potentially unsafe and harmful responses from a language model, and it's an option of the 'AI\_COMPLETE (or 'SNOWFLAKE.CORTEX.COMPLETE) function. Option B is correct as 'COMPLETE (SNOWFLAKE.CORTEX)' supports the 'guardrails' argument. Option C is correct as the Cortex REST API endpoint `7api/v2/cortex/inference:complete` accepts 'guardrails' as an optional JSON argument. Option D is correct as the Cortex Playground allows users to 'Enable Cortex Guard' to implement safeguards. option E is correct because ' TRY\_COMPLETE (SNOWFLAKE.CORTEX) performs the same operation as 'COMPLETE and also supports the 'guardrails' argument. Option A is incorrect because 'CLASSIFY TEXT is a task-specific function and does not have the 'guardrails' option; Cortex Guard is associated with generative completion functions like 'COMPLETE.

#### NEW QUESTION # 54

.....

On the one hand, GES-C01 test torrent is revised and updated according to the changes in the syllabus and the latest developments



rvViD\_XFN5qd9flSy6PefbT8NFbwRUM