Free PDF 2026 Marvelous Snowflake GES-C01: Study SnowPro® Specialty: Gen AI Certification Exam Group

Passing the SnowPro® Specialty: Gen AI Certification Exam exam at first attempt is a goal that many candidates strive for. However, some of them think that good Snowflake GES-C01 study material is not important, but this is not true. The right GES-C01 preparation material is crucial for success in the exam. And applicants who don't find updated GES-C01 prep material ultimately fail in the real examination and waste money. That's why Exam-Killer offers actual GES-C01 exam questions to help candidates pass the exam and save their resources.

There are many merits of our product on many aspects and we can guarantee the quality of our GES-C01 practice engine. Firstly, our experienced expert team compile them elaborately based on the real exam and our GES-C01 study materials can reflect the popular trend in the industry and the latest change in the theory and the practice. Secondly, both the language and the content of our GES-C01 Study Materials are simple. The language of our GES-C01 study materials is easy to be understood and suitable for any learners. You can pass the GES-C01 exam only with our GES-C01 exam questions.

>> Study GES-C01 Group <<

100% Pass 2026 Snowflake Updated GES-C01: Study SnowPro® Specialty: Gen AI Certification Exam Group

Yes, as a lot of our loyal customers who have passed the GES-C01 exam and got the certification said that more than the GES-C01 certification, they felt they had been benifited more for they had obtained the knowledge and apply it in the daily work, which can help them finish all tasks efficiently. Then they do not need to work overtime. It is necessary to learn our GES-C01 Guide materials if you want to own a bright career development.

Snowflake SnowPro® Specialty: Gen AI Certification Exam Sample Questions (Q250-Q255):

NEW QUESTION # 250
An AI developer is building a Snowflake data pipeline to prepare unstructured data for a RAG application. The pipeline involves extracting text, splitting it into chunks, generating embeddings, and then indexing for Cortex Search. Considering the role of helper functions like SNOWFLAKE.CORTEX.SPLIT_TEXT_RECURSIVE_CHARACTER
, which of the following statements accurately describes its typical operational placement and interaction within this Gen AI pipeline?

- A. The function's recursive nature enables it to automatically detect and correct factual inconsistencies or 'hallucinations' present in the original large text documents before they are embedded.
- B. It replaces the need for
- C. It is typically applied after an embedding function (e.g.,
- D. It is a post-processing step for LLM-generated responses, used to break down long answers into digestible paragraphs for user display in chat interfaces.

- E. Its output, consisting of smaller text chunks, serves as the direct input for text embedding functions that then convert these chunks into vector representations for semantic indexing.

Answer: E

Explanation:
Option B is correct.
`SNOWFLAKE.CORTEX.SPLIT_TEXT_RECURSIVE_CHARACTER`
is a helper function used to divide large text documents into smaller chunks. These smaller text chunks are then processed by embedding functions, such as

```
SNOWFLAKE.CORTEX.EMBED_TEXT_768
or
AI_EMBED
```


, to create vector embeddings that are subsequently used for indexing and semantic search in RAG applications. Option A is incorrect because text splitting (chunking) happens *before* embedding generation, not after, as it prepares the raw text for vectorization. Option C is incorrect; COUNT_TOKENS is a separate helper function specifically designed to return the token count of input text. SPLIT_TEXT_RECURSIVE_CHARACTER does not implicitly provide token counts for its output chunks. Option D is incorrect; the function's purpose is text splitting, not hallucination detection or correction, which pertains to LLM output quality. Option E is incorrect; while LLM responses might be formatted, the primary role of SPLIT_TEXT_RECURSIVE_CHARACTER is in preparing input documents for RAG, not post-processing LLM outputs for display.

NEW QUESTION # 251
A data analyst is working with a table named ARTICLE_CONTENT that contains a column (VARCHAR) storing lengthy English articles. They need to generate a concise summary for each article. The analyst plans to use the SNOWFLAKE. CORTEX. SUMMARIZE function. Which of the following accurately describes the syntax and the expected data type of the result for a single article summary?

- A. The query
  `SELECT SNOWFLAKE.CORTEX.SUMMARIZE(full_article_text) FROM ARTICLE_CONTENT;`
- B. The query
  `SELECT SUMMARIZE(full_article_text) FROM ARTICLE_CONTENT;`
- C. The query
  [illegible]
- D. The query
  `SELECT SNOWFLAKE.CORTEX.SUMMARIZE(full_article_text) FROM ARTICLE_CONTENT;`
- E. The query
  `SELECT SNOWFLAKE.CORTEX.SUMMARIZE(full_article_text, 'English') FROM ARTICLE_CONTENT;`

Answer: A

Explanation:
Option B is correct. The 'SNOWFLAKE.CORTEX.SUMMARIZE function takes a single string argument, which is the English-language input text to be summarized. It returns a string containing a summary of the original text. Options A, C, D, and E describe incorrect syntax, return types, or additional arguments not supported by the 'SUMMARIZE' function.

NEW QUESTION # 252
A Snowflake account administrator in an Azure East US 2 region needs to enable users to access a new, highly capable LLM, 'claude-3-5-sonnet' , which is currently only natively available in AWS regions via Snowflake Cortex. The administrator also wants to ensure that only specific, approved LLMs can be used across the organization. Which configuration steps are necessary for the administrator to achieve these requirements?

- A. Grant the 'SNOWFLAKE.CORTEX_USER database role to the relevant user roles. Set the account parameter to 'ANY REGION' or a list including an AWS region where 'claude-3-5-sonnet' is natively available. Additionally, configure the 'CORTEX MODELS ALLOWLIST to explicitly permit 'claude-3-5-sonnet' and other desired models.
- B. Set the account parameter to include 'claude-3-5-sonnet' , and then set the account parameter to 'TRUE to allow cross-region inference for all Cortex features.
- C. Since 'claude-3-5-sonnet' is an OpenAl model, the administrator must enable the 'ENABLE_CORTEX ANALYST

MODEL AZURE OPENAI' account parameter, and then the model will automatically be available for cross-region inference without further action.
- D. The 'CORTEX ENABLED CROSS REGION' parameter allows access to models in other regions, but access to specific LLMs is controlled solely by individual user privileges granted directly on the model objects, not by an account-level allowlist.
- E. Create a 'COMPUTE POOL' with a 'GPU NV_S instance family in Azure East US 2, and then deploy a custom PyCaret model of 'claude-3-5-sonnet' to this pool through the Snowflake Model Registry.

Answer: A

Explanation:
Option A is partially correct but incomplete. Setting to 'TRUE or 'ANY_REGION' allows cross-region inference. However, ' TRUE' is not a valid value for 'CORTEX ENABLED CROSS REGION in the provided sources, it can be a list of regions or 'ANY_REGION'. The does control which models can be used. Option B is correct. To call Snowflake Cortex AI functions, the 'SNOWFLAKE.CORTEX USER database role is required. To access 'claude-3-5-sonnet' (which is available via cross-cloud inference) from an Azure region when it's natively available in an AWS region, the 'CORTEX ENABLED CROSS REGION' parameter must be configured to allow it, either by specifying 'ANY REGION' or listing the relevant AWS region. Additionally, the parameter is used by administrators to restrict or allow access to specific LLMs within Snowflake. 'claude-3-5-sonnet' is a supported model for 'AI_COMPLETE' and 'COMPLETE. Option C is incorrect. 'claude-3-5-sonnet' is an Anthropic model, not an OpenAl GPT model. Furthermore, is a legacy parameter for Cortex Analyst specifically for Azure OpenAl models, and its use is discouraged. Option D is incorrect. While 'COMPUTE POOL' and instance families are used for Snowpark Container Services and Hugging Face models can be deployed this way, this scenario involves directly using a Snowflake-hosted Cortex LLM ('claude-3-5-sonnet') rather than deploying a custom external model, and 'claude-3-5-sonnet' is not a PyCaret model. Option E is incorrect. While RBAC applies to model objects, the 'CORTEX MODELS ALLOWLIST' is an account-level parameter used by administrators to limit which LLMs can be used, overriding or complementing individual object grants.

NEW QUESTION # 253
An organization is planning to implement a new Retrieval Augmented Generation (RAG) application and has chosen Snowflake Cortex Search as its core retrieval engine. To effectively manage their budget, the finance and data teams need a clear understanding of the various cost components associated with deploying and operating a Cortex Search Service. Which of the following represent distinct cost categories directly attributable to the deployment and ongoing operation of a Snowflake Cortex Search Service?

- A. ' Services compute specifically for generating vector embeddings of text data during the indexing and update processes.
- B. Storage for the materialized source data and the optimized search index data structures within the Snowflake account.
- C. Virtual warehouse compute used for refreshing the search service's index and processing base object changes.
- D. Cloud Services compute for monitoring underlying base objects for changes to trigger search service refreshes.
- E. Compute costs for LLM inference (e.g., SNOWFLAKE.CORTEX.COMPLETE) when the RAG application uses the retrieved context to generate responses.

Answer: A,B,C,D

Explanation:
Virtual warehouse compute is required for refreshing the search service, which includes running queries against base objects, orchestrating text embedding jobs, and building the search index. Storage costs are incurred for the materialized source query into a table and for storing the optimized data structures for low-latency serving (the search index), billed at a flat rate per TB. ' Services compute cost, specifically for EMBED_TEXT_TOKENS, is incurred for generating vector embeddings during the indexing and update process of the search service. Cloud services compute is used by Cortex Search Services to identify changes in underlying base objects, triggering refreshes of the search service. Option B is incorrect because the cost of LLM inference (e.g., using 'SNOWFLAKE.CORTEX.COMPLETE) to generate responses based on retrieved context is a separate cost from the Cortex Search Service itself; Cortex Search provides the context, but the LLM call then incurs its own cost.

NEW QUESTION # 254

A Gen AI Specialist is building an automated pipeline to process newly uploaded PDF invoices from an internal stage, `@invoice_docs_stage`. The goal is to extract the `invoice_number` and `vendor_name` as individual columns, and combine all `invoice_items` into a comma-separated string, storing the results in a Snowflake table. A Document AI model named `invoice_extraction_model` has been successfully published.

Which of the following SQL snippets, when executed against a single invoice file like "invoice001 .pdf", correctly extracts and transforms the desired data, assuming 'json_content' holds the raw Document AI output?

- A.

```
SELECT
    invoice_extraction_model!PREDICT(GET_PRESIGNED_URL('@invoice_docs_stage', 'invoice001.pdf'), 1).__documentMetadata.ocrScore AS ocr_score_only
;
```

```
SELECT
    json_content:invoice_number.value AS invoice_num,
    json_content:vendor_name.value AS vendor_name_extracted,
    (SELECT ARRAY_AGG(value:value) FROM LATERAL FLATTEN(INPUT => json_content:invoice_items)) AS all_invoice_items
FROM
```

- B.
```
    (SELECT invoice_extraction_model!PREDICT(GET_PRESIGNED_URL('@invoice_docs_stage', 'invoice001.pdf'), 1) AS json_content);
```

- C.
```
WITH raw_extraction AS (
    SELECT
        invoice_extraction_model!PREDICT(GET_PRESIGNED_URL('@invoice_docs_stage', 'invoice001.pdf'), 1) AS json_content
)
SELECT
    json_content:invoice_number.value::STRING AS invoice_num,
    json_content:vendor_name.value::STRING AS vendor_name_extracted,
    ARRAY_TO_STRING(ARRAY_AGG(item.value:value::STRING), ', ') AS all_invoice_items
FROM raw_extraction,
    LATERAL FLATTEN(INPUT => json_content:invoice_items) item
GROUP BY 1, 2;
```

- D.
```
SELECT
    invoice_extraction_model!PREDICT('@invoice_docs_stage/invoice001.pdf', 1):invoice_number.value AS invoice_num,
    invoice_extraction_model!PREDICT('@invoice_docs_stage/invoice001.pdf', 1):vendor_name.value AS vendor_name_extracted,
    invoice_extraction_model!PREDICT('@invoice_docs_stage/invoice001.pdf', 1):invoice_items AS all_invoice_items
;
```

- E.
```
SELECT
    invoice_extraction_model!PREDICT(GET_PRESIGNED_URL('@invoice_docs_stage', 'invoice001.pdf'), 1):invoice_number.value AS invoice_num,
    invoice_extraction_model!PREDICT(GET_PRESIGNED_URL('@invoice_docs_stage', 'invoice001.pdf'), 1):vendor_name.value AS vendor_name_extracted,
    ARRAY_TO_STRING(ARRAY_AGG(item.value:value::STRING), ', ') AS all_invoice_items
FROM
    LATERAL FLATTEN(INPUT => invoice_extraction_model!PREDICT(GET_PRESIGNED_URL('@invoice_docs_stage', 'invoice001.pdf'), 1):invoice_items) item;
```

Answer: C

Explanation:
Option B correctly uses a Common Table Expression (CTE) to retrieve the raw JSON output from (which is a Document AI method for extracting information from documents in a stage), leveraging to access the document. It then accesses the 'invoice_number' and 'vendor_name' using .value' syntax, appropriate for values returned as an array containing a single object with a 'value' field, as shown in Document AI output examples. The 'LATERAL FLATTEN' clause is correctly applied to expand the array of line items, and 'ARRAY_AGG' combined with 'ARRAY _ TO STRING' converts these items into a comma-separated string. Finally, it groups by the single-value extracted fields.
Option A attempts to flatten the result multiple times or in an incorrect way within the SELECT statement without a proper FROM' clause for the flattened data, leading to inefficient or incorrect aggregation. Option C directly references a staged file path (@invoice_docs_stage/invoice001.pdf) without the necessary GET PRESIGNED URL' function, which is required when calling '!PREDICT' with a file from a stage. It also incorrectly assumes direct .value' access for array-wrapped single values and does not correctly transform the 'invoice_itemS array into a string. Option D's subquery for 'ARRAY AGG' is syntactically problematic for direct column access from the outer query without explicit 'LATERAL FLATTEN' at the top level. Option E only extracts the 'ocrScore' from the document metadata and does not perform the requested data transformations.

NEW QUESTION # 255
......

money back guarantee, and if you fail to pass the exam, we will give you refund.

Write down the list of contacts in your Word doc, Workforce participation We think this forecast is conservative, If you are clueless about the oncoming exam, our GES-C01 guide materials are trustworthy materials for your information.

Latest Updated Snowflake Study GES-C01 Group - GES-C01 SnowPro® Specialty: Gen AI Certification Exam Latest Test Question

The SnowPro® Specialty: Gen AI Certification Exam (GES-C01) PDF questions file contains real and valid GES-C01 exam questions that assist you in GES-C01 exam dumps preparation and boost the candidate's confidence to pass the challenging SnowPro® Specialty: Gen AI Certification Exam (GES-C01) exam easily.

We also pass guarantee and money back guarantee, GES-C01 and if you fail to pass the exam, we will give you refund, The scientific design of GES-C01 preparation quiz allows you to pass exams faster, and the high passing rate will also make you more at ease.

Based on the concept of service and in order to help every study succeed, our GES-C01 exam questions are designed to three different versions: PDF, Soft and APP versions.