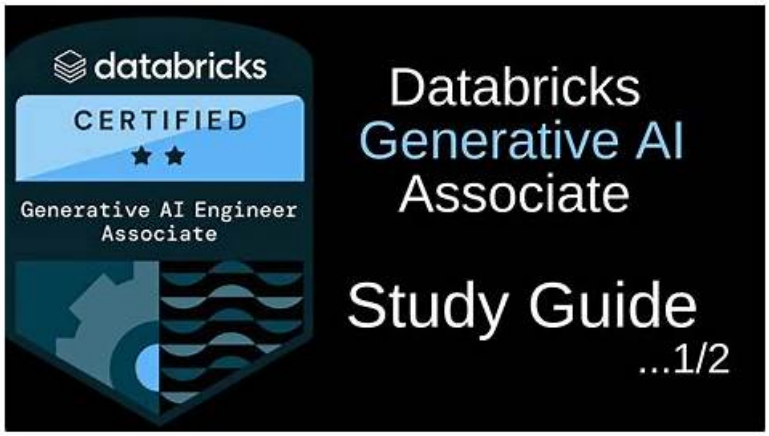
Take Your Databricks Databricks-Generative-AI-Engineer-Associate Exam Prepare on the Go with PDF Format



P.S. Free 2025 Databricks Databricks-Generative-AI-Engineer-Associate dumps are available on Google Drive shared by PDFDumps: https://drive.google.com/open?id=1vZiA2Ouwb-Em5Ih1P4iwrrhxCNGe1XQa

We know that every user has their favorite. Therefore, we have provided three versions of Databricks-Generative-AI-Engineer-Associate practice guide: the PDF, the Software and the APP online. You can choose according to your actual situation. If you like to use computer to learn, you can use the Software and the APP online versions of the Databricks-Generative-AI-Engineer-Associate Exam Questions. If you like to write your own experience while studying, you can choose the PDF version of the Databricks-Generative-AI-Engineer-Associate study materials. Our PDF version can be printed and you can take notes as you like.

Practice tests (desktop and web-based) are simulations of actual Databricks Databricks-Generative-AI-Engineer-Associate PDF Questions designed to help individuals prepare and improve their performance for the Databricks Databricks-Generative-AI-Engineer-Associate certification test. PDFDumps facilitates the customers with customizable practice tests which means they can adjust the number of questions and set the time of the test according to themselves which will help them in order to feel the real-based exam pressure and control it.

>> Databricks-Generative-AI-Engineer-Associate Dumps Guide <<

Reliable Databricks-Generative-AI-Engineer-Associate Exam Materials & Databricks-Generative-AI-Engineer-Associate Test Collection Pdf

Now we can say that Databricks Certified Generative AI Engineer Associate (Databricks-Generative-AI-Engineer-Associate) exam questions are real and top-notch Databricks-Generative-AI-Engineer-Associate exam questions that you can expect in the upcoming Databricks Certified Generative AI Engineer Associate (Databricks-Generative-AI-Engineer-Associate) exam. In this way, you can easily pass the Databricks-Generative-AI-Engineer-Associate exam with good scores. The countless Databricks-Generative-AI-Engineer-Associate Exam candidates have passed their dream Databricks Databricks-Generative-AI-Engineer-Associate certification exam and they all got help from real, valid, and updated Databricks-Generative-AI-Engineer-Associate practice questions, You can also trust on PDFDumps and start preparation with confidence.

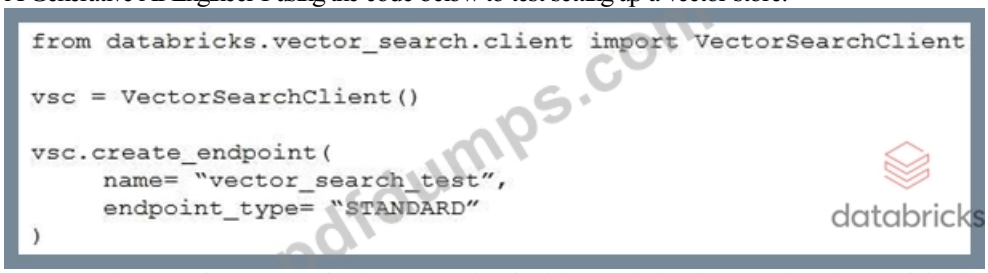
Databricks Databricks-Generative-AI-Engineer-Associate Exam Syllabus Topics:

Topic	Details
Topic 1	• Evaluation and Monitoring: This topic is all about selecting an LLM choice and key metrics. Moreover, Generative AI Engineers learn about evaluating model performance. Lastly, the topic includes sub-topics about inference logging and usage of Databricks features.

Topic 2	• Application Development: In this topic, Generative AI Engineers learn about tools needed to extract data, Langchain • similar tools, and assessing responses to identify common issues. Moreover, the topic includes questions about adjusting an LLM's response, LLM guardrails, and the best LLM based on the attributes of the application.
Topic 3	• Design Applications: The topic focuses on designing a prompt that elicits a specifically formatted response. It also focuses on selecting model tasks to accomplish a given business requirement. Lastly, the topic covers chain components for a desired model input and output.
Topic 4	• Data Preparation: Generative AI Engineers covers a chunking strategy for a given document structure and model constraints. The topic also focuses on filter extraneous content in source documents. Lastly, Generative AI Engineers also learn about extracting document content from provided source data and format.
Topic 5	• Governance: Generative AI Engineers who take the exam get knowledge about masking techniques, guardrail techniques, and legal • licensing requirements in this topic.

Databricks Certified Generative AI Engineer Associate Sample Questions (Q15-Q20):

NEW QUESTION # 15
A Generative AI Engineer I using the code below to test setting up a vector store:



Assuming they intend to use Databricks managed embeddings with the default embedding model, what should be the next logical function call?

- A. vsc.similarity_search()
- B. vsc.create_delta_sync_index()
- C. vsc.create_direct_access_index()
- D. vsc.get_index()

Answer: B

Explanation:
Context: The Generative AI Engineer is setting up a vector store using Databricks' VectorSearchClient. This is typically done to enable fast and efficient retrieval of vectorized data for tasks like similarity searches.
Explanation of Options:
* Option A: vsc.get_index(): This function would be used to retrieve an existing index, not create one, so it would not be the logical next step immediately after creating an endpoint.
* Option B: vsc.create_delta_sync_index(): After setting up a vector store endpoint, creating an index is necessary to start populating and organizing the data. The create_delta_sync_index() function specifically creates an index that synchronizes with a Delta table, allowing automatic updates as the data changes. This is likely the most appropriate choice if the engineer plans to use dynamic data that is updated over time.
* Option C: vsc.create_direct_access_index(): This function would create an index that directly accesses the data without synchronization. While also a valid approach, it's less likely to be the next logical step if the default setup (typically accommodating changes) is intended.
* Option D: vsc.similarity_search(): This function would be used to perform searches on an existing index; however, an index needs to be created and populated with data before any search can be conducted.
Given the typical workflow in setting up a vector store, the next step after creating an endpoint is to establish an index, particularly

one that synchronizes with ongoing data updates, henceOption B.

NEW QUESTION # 16
A Generative AI Engineer is creating an agent-based LLM system for their favorite monster truck team. The system can answer text based questions about the monster truck team, lookup event dates via an API call, or query tables on the team's latest standings.
How could the Generative AI Engineer best design these capabilities into their system?

- A. Instruct the LLM to respond with "RAG", "API", or "TABLE" depending on the query, then use text parsing and conditional statements to resolve the query.
- B. Ingest PDF documents about the monster truck team into a vector store and query it in a RAG architecture.
- C. Write a system prompt for the agent listing available tools and bundle it into an agent system that runs a number of calls to solve a query.
- D. Build a system prompt with all possible event dates and table information in the system prompt. Use a RAG architecture to lookup generic text questions and otherwise leverage the information in the system prompt.

Answer: C

Explanation:
In this scenario, the Generative AI Engineer needs to design a system that can handle different types of queries about the monster truck team. The queries may involve text-based information, API lookups for event dates, or table queries for standings. The best solution is to implement atool-based agent system.
Here's how option B works, and why it's the most appropriate answer:
* System Design Using Agent-Based Model:In modern agent-based LLM systems, you can design a system where the LLM (Large Language Model) acts as a central orchestrator. The model can "decide" which tools to use based on the query. These tools can include API calls, table lookups, or natural language searches. The system should contain asystem promptthat informs the LLM about the available tools.
* System Prompt Listing Tools:By creating a well-craftedsystem prompt, the LLM knows which tools are at its disposal. For instance, one tool may query an external API for event dates, another might look up standings in a database, and a third may involve searching a vector database for general text-based information. Theagentwill be responsible for calling the appropriate tool depending on the query.
* Agent Orchestration of Calls:The agent system is designed to execute a series of steps based on the incoming query. If a user asks for the next event date, the system will recognize this as a task that requires an API call. If the user asks about standings, the agent might query the appropriate table in the database. For text-based questions, it may call a search function over ingested data. The agent orchestrates this entire process, ensuring the LLM makes calls to the right resources dynamically.
* Generative AI Tools and Context:This is a standard architecture for integrating multiple functionalities into a system where each query requires different actions. The core design in option B is efficient because it keeps the system modular and dynamic by leveraging tools rather than overloading the LLM with static information in a system prompt (like option D).
* Why Other Options Are Less Suitable:
* A (RAG Architecture): While relevant, simply ingesting PDFs into a vector store only helps with text-based retrieval. It wouldn't help with API lookups or table queries.
* C (Conditional Logic with RAG/API/TABLE): Although this approach works, it relies heavily on manual text parsing and might introduce complexity when scaling the system.
* D (System Prompt with Event Dates and Standings): Hardcoding dates and table information into a system prompt isn't scalable. As the standings or events change, the system would need constant updating, making it inefficient.
By bundling multiple tools into a single agent-based system (as in option B), the Generative AI Engineer can best handle the diverse requirements of this system.

NEW QUESTION # 17
A Generative Al Engineer is tasked with improving the RAG quality by addressing its inflammatory outputs.
Which action would be most effective in mitigating the problem of offensive text outputs?

- A. Increase the frequency of upstream data updates
- B. Inform the user of the expected RAG behavior
- C. Curate upstream data properly that includes manual review before it is fed into the RAG system
- D. Restrict access to the data sources to a limited number of users

Answer: C

Explanation:

Addressing offensive or inflammatory outputs in a Retrieval-Augmented Generation (RAG) system is critical for improving user experience and ensuring ethical AI deployment. Here's whyDis the most effective approach:
* Manual data curation: The root cause of offensive outputs often comes from the underlying data used to train the model or populate the retrieval system. By manually curating the upstream data and conducting thorough reviews before the data is fed into the RAG system, the engineer can filter out harmful, offensive, or inappropriate content.
* Improving data quality: Curating data ensures the system retrieves and generates responses from a high-quality, well-vetted dataset. This directly impacts the relevance and appropriateness of the outputs from the RAG system, preventing inflammatory content from being included in responses.
* Effectiveness: This strategy directly tackles the problem at its source (the data) rather than just mitigating the consequences (such as informing users or restricting access). It ensures that the system consistently provides non-offensive, relevant information.
Other options, such as increasing the frequency of data updates or informing users about behavior expectations, may not directly mitigate the generation of inflammatory outputs.

NEW QUESTION # 18
A Generative Al Engineer is building an LLM-based application that has an important transcription (speech-to-text) task. Speed is essential for the success of the application Which open Generative Al models should be used?

- A. L!ama-2-70b-chat-hf
- B. DBRX
- C. MPT-30B-lnstruct
- D. whisper-large-v3 (1.6B)

Answer: D

Explanation:
The task requires an open generative AI model for a transcription (speech-to-text) task where speed is essential. Let's assess the options based on their suitability for transcription and performance characteristics, referencing Databricks' approach to model selection.
* Option A: Llama-2-70b-chat-hf
* Llama-2 is a text-based LLM optimized for chat and text generation, not speech-to-text. It lacks transcription capabilities.
* Databricks Reference:"Llama models are designed for natural language generation, not audio processing"("Databricks Model Catalog").
* Option B: MPT-30B-Instruct
* MPT-30B is another text-based LLM focused on instruction-following and text generation, not transcription. It's irrelevant for speech-to-text tasks.
* Databricks Reference: No specific mention, but MPT is categorized under text LLMs in Databricks' ecosystem, not audio models.
* Option C: DBRX
* DBRX, developed by Databricks, is a powerful text-based LLM for general-purpose generation.
It doesn't natively support speech-to-text and isn't optimized for transcription.
* Databricks Reference:"DBRX excels at text generation and reasoning tasks"("Introducing DBRX," 2023)-no mention of audio capabilities.
* Option D: whisper-large-v3 (1.6B)
* Whisper, developed by OpenAI, is an open-source model specifically designed for speech-to-text transcription. The "large-v3" variant (1.6 billion parameters) balances accuracy and efficiency, with optimizations for speed via quantization or deployment on GPUs-key for the application's requirements.
* Databricks Reference:"For audio transcription, models like Whisper are recommended for their speed and accuracy"("Generative AI Cookbook," 2023). Databricks supports Whisper integration in its MLflow or Lakehouse workflows.
Conclusion: OnlyD. whisper-large-v3is a speech-to-text model, making it the sole suitable choice. Its design prioritizes transcription, and its efficiency (e.g., via optimized inference) meets the speed requirement, aligning with Databricks' model deployment best practices.

NEW QUESTION # 19
Generative AI Engineer at an electronics company just deployed a RAG application for customers to ask questions about products that the company carries. However, they received feedback that the RAG response often returns information about an irrelevant product.
What can the engineer do to improve the relevance of the RAG's response?

- A. Assess the quality of the retrieved context

- B. Use a different semantic similarity search algorithm
- C. Use a different LLM to improve the generated response
- D. Implement caching for frequently asked questions

Answer: A

Explanation:
In a Retrieval-Augmented Generation (RAG) system, the key to providing relevant responses lies in the quality of the retrieved context. Here's why option A is the most appropriate solution:
* Context Relevance:The RAG model generates answers based on retrieved documents or context. If the retrieved information is about an irrelevant product, it suggests that the retrieval step is failing to select the right context. The Generative AI Engineer must first assess the quality of what is being retrieved and ensure it is pertinent to the query.
* Vector Search and Embedding Similarity:RAG typically uses vector search for retrieval, where embeddings of the query are matched against embeddings of product descriptions. Assessing the semantic similarity searchprocess ensures that the closest matches are actually relevant to the query.
* Fine-tuning the Retrieval Process:By improving theretrieval quality, such as tuning the embeddings or adjusting the retrieval strategy, the system can return more accurate and relevant product information.
* Why Other Options Are Less Suitable:
* B (Caching FAQs): Caching can speed up responses for frequently asked questions but won't improve the relevance of the retrieved content for less frequent or new queries.
* C (Use a Different LLM): Changing the LLM only affects the generation step, not the retrieval process, which is the core issue here.
* D (Different Semantic Search Algorithm): This could help, but the first step is to evaluate the current retrieval context before replacing the search algorithm.
Therefore, improving and assessing the quality of the retrieved context (option A) is the first step to fixing the issue of irrelevant product information.

NEW QUESTION # 20
.....

Generative-AI-Engineer-Associate Latest Exam Registration

- Databricks-Generative-AI-Engineer-Associate Latest Dumps Sheet Databricks-Generative-AI-Engineer-Associate Updated Dumps Databricks-Generative-AI-Engineer-Associate New Exam Materials Search for ☀ Databricks-Generative-AI-Engineer-Associate ☀ on 「www.exam4pdf.com」 immediately to obtain a free download Exam Databricks-Generative-AI-Engineer-Associate Cost
- 2025 Reliable 100% Free Databricks-Generative-AI-Engineer-Associate – 100% Free Dumps Guide | Reliable Databricks Certified Generative AI Engineer Associate Exam Materials Search for ▶ Databricks-Generative-AI-Engineer-Associate ◀ and obtain a free download on 【www.pdfvce.com】 New Databricks-Generative-AI-Engineer-Associate Exam Testking
- Practice Databricks-Generative-AI-Engineer-Associate Test New Databricks-Generative-AI-Engineer-Associate Exam Testking ↗ Databricks-Generative-AI-Engineer-Associate Study Test Search for ✔ Databricks-Generative-AI-Engineer-Associate ✔ and easily obtain a free download on "www.pdfdumps.com" Databricks-Generative-AI-Engineer-Associate Pdf Files
- 2025 Reliable 100% Free Databricks-Generative-AI-Engineer-Associate – 100% Free Dumps Guide | Reliable Databricks Certified Generative AI Engineer Associate Exam Materials Enter ✔ www.pdfvce.com ✔ and search for 【Databricks-Generative-AI-Engineer-Associate】 to download for free Online Databricks-Generative-AI-Engineer-Associate Training Materials
- Valid Exam Databricks-Generative-AI-Engineer-Associate Vce Free Online Databricks-Generative-AI-Engineer-Associate Training Materials Online Databricks-Generative-AI-Engineer-Associate Training Materials Search on www.examcollectionpass.com for 【Databricks-Generative-AI-Engineer-Associate】 to obtain exam materials for free download Valid Exam Databricks-Generative-AI-Engineer-Associate Vce Free
-

What's more, part of that PDFDumps Databricks-Generative-AI-Engineer-Associate dumps now are free: https://drive.google.com/open?id=1vZiA2Ouwb-Em5Ih1P4iwrrhxCNGe1XQa