DUMPSBOSS.

Oracle Cloud Infrastructure 2026 Generative AI Professional

Oracle 1z0-1127-25

Version Demo

Total Demo Questions: 9

Total Premium Questions: 95

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1

What is the primary function of the " temperature " parameter in the OCI Generative AI Generation models?

A. Controls the randomness of the model ' s output, affecting its creativity

B. Specifies a string that tells the model to stop generating more content

C. Assigns a penalty to tokens that have already appeared in the preceding text

D. Determines the maximum number of tokens the model can generate per response

ANSWER: A

Explanation:

Controls the randomness of the model ' s output, affecting its creativity is correct because temperature is an inference parameter that controls how broadly the generation model samples from its available next-token choices. A lower temperature concentrates selection on the most likely tokens, producing more consistent, focused, and repeatable responses. A higher temperature distributes probability more broadly across possible tokens, increasing variation and enabling more imaginative or diverse phrasing. This makes temperature an important control for matching output behavior to the task: lower settings are generally suitable when predictability and precision are important, while higher settings can be useful for ideation, creative drafting, or generating alternative formulations.

In OCI Generative AI, temperature is supplied as part of the model inference configuration for supported text-generation models. It influences the sampling behavior during generation; it does not alter the model's training, knowledge base, or the user's prompt. Oracle documents generation and inference configuration for pretrained models in its Generative AI service documentation. See Oracle Cloud Infrastructure Generative AI: Pretrained Models and Oracle Cloud Infrastructure Generative AI: Inferencing.

QUESTION NO: 2

Accuracy in vector databases contributes to the effectiveness of Large Language Models (LLMs) by preserving a specific type of relationship. What is the nature of these relationships, and why arethey crucial for language models?

A. Linear relationships; they simplify the modeling process

B. Semantic relationships; crucial for understanding context and generating precise language

C. Hierarchical relationships; important for structuring database queries

D. Temporal relationships; necessary for predicting future linguistic trends

ANSWER: B

Explanation:

Semantic relationships; crucial for understanding context and generating precise language is correct because vector databases store embeddings: numerical representations of text whose locations in a high-dimensional vector space reflect meaning. Content with similar concepts, intent, or contextual usage is positioned nearer together, even when it does not use the same words. Accurate embedding storage and similarity search therefore allow an LLM application to retrieve passages that are meaningfully relevant to a user's request.

This is especially important in retrieval-augmented generation workflows. Rather than relying only on information encoded during model training, the application converts a query into an embedding, finds semantically similar document chunks, and supplies those results as grounded context to the model. Preserving semantic proximity improves the relevance of retrieved evidence, helping the model interpret the request in context and produce more precise, useful responses. Oracle describes AI Vector Search as using vector embeddings to represent the semantic meaning of unstructured data and to identify similar content efficiently. Oracle's retrieval-augmented generation guidance likewise describes grounding generative AI responses with retrieved enterprise knowledge. See Oracle AI Vector Search overview and Oracle Retrieval-Augmented Generation documentation.

QUESTION NO: 3

A company needs a customized model that consistently produces incident reports in a mandated internal format. The report format is stable, the company has many high-quality input-and-output examples, and the required behavior cannot be expressed reliably with a short prompt. Which customization method is most appropriate?

A. Fine-tune a supported foundation model using the representative incident-report examples.

B. Use RAG as the only customization method for the fixed report format.

C. Increase temperature so that the model produces more varied report structures.

D. Store example reports in a vector database without providing task-specific training examples.

ANSWER: A

Explanation:

Fine-tuning is appropriate because the organization needs a stable, repeatable behavioral adaptation and has a substantial set of representative labeled examples. The customized model can learn the expected mapping between incident details and the mandated report format, reducing the need to include extensive demonstrations in every inference request.

RAG is intended primarily to supply external factual context, especially information that changes over time. A higher temperature increases variation and is contrary to a fixed format. A vector database can retrieve similar reports, but retrieval alone does not train the model to follow a required output structure consistently.

QUESTION NO: 4

A RAG assistant answers a question correctly when a policy document is retrieved, but it occasionally states additional policy details that are absent from every retrieved passage. The answer remains fluent and directly addresses the question. Which evaluation dimension is most clearly deficient?

A. Groundedness

B. Answer relevance

C. Tokenization

D. Embedding dimensionality

ANSWER: A

Explanation:

Groundedness is deficient because the additional claims are not supported by the retrieved context. Answer relevance can still be high when the response addresses the question directly, and fluency concerns readability rather than evidence support. A grounded RAG response should base its factual claims on the retrieved information supplied for that request.

QUESTION NO: 5

A team uses OCI Generative AI to produce short, structured classifications from a fixed set of categories. The same request should produce highly consistent results, and creative variation is undesirable. Which decoding configuration best supports this requirement?

A. Use a low temperature and restrictive sampling settings.

B. Use a high temperature and a broad top-p setting.

C. Use a high presence penalty to maximize vocabulary diversity.

D. Use a larger maximum output-token value to make classifications more deterministic.

ANSWER: A

Explanation:

A low temperature with restrictive sampling supports consistent output because it concentrates selection on the highest-probability tokens. For a structured classification task, the goal is repeatability rather than diversity.

High temperature broadens the probability distribution and increases variation. A large top-p setting permits a broader candidate set, which is generally less suitable when deterministic behavior is preferred. A presence penalty encourages introducing tokens that have not yet appeared and is not a control for classification consistency.

QUESTION NO: 6

Which statement is true about the " Top p " parameter of the OCI Generative AI Generation models?

A. " Top p " selects tokens from the " Top k " tokens sorted by probability.

B. " Top p " assigns penalties to frequently occurring tokens.

C. " Top p " limits token selection based on the sum of their probabilities.

D. " Top p " determines the maximum number of tokens per response.

ANSWER: C

Explanation:

"Top p" limits token selection based on the sum of their probabilities. This sampling control is also known as nucleus sampling. At each step of generation, the model ranks possible next tokens by probability and forms the smallest candidate set whose cumulative probability reaches the configured top-p threshold. The next token is then sampled from that probability mass. For example, a lower threshold restricts sampling to a narrower set of highly likely tokens, generally producing more focused and predictable output. A higher threshold allows a broader set of plausible tokens to participate, which can increase variation in the generated response.

In OCI Generative AI inference requests, top-p is a decoding parameter used alongside controls such as temperature and maximum output tokens. It affects how the model chooses each successive token; it does not define the response length. Choosing an appropriate value depends on the desired balance between consistency and creativity for the application. Oracle documents generation inference request parameters in its API reference and provides OCI Generative AI service guidance at Oracle Cloud Infrastructure Generative AI Inference API and OCI Generative AI documentation.

QUESTION NO: 7

How does the structure of vector databases differ from traditional relational databases?

A. It stores data in a linear or tabular format.

B. It is not optimized for high-dimensional spaces.

C. It uses simple row-based data storage.

D. It is based on distances and similarities in a vector space.

ANSWER: D

Explanation:

It is based on distances and similarities in a vector space. Vector databases represent information such as text, images, or other content as numerical embedding vectors: ordered arrays of numbers that encode semantic characteristics. Rather than primarily retrieving records through exact values, predefined columns, and relational joins, a vector database indexes these vectors to efficiently identify nearby vectors in a high-dimensional space. “Near” is calculated using similarity or distance

measures, commonly cosine similarity, Euclidean distance, or dot product. This structure makes vector storage especially useful for semantic retrieval, where content with similar meaning can be returned even when it does not contain the same keywords as a query. In retrieval-augmented generation workflows, an application embeds a user query, performs a vector similarity search over embedded enterprise content, and supplies the most relevant retrieved passages to a large language model. Oracle Database supports vector data types and vector indexes for this purpose, enabling AI Vector Search to find data based on semantic similarity. Oracle describes vector search as finding vectors that are similar to a query vector according to a distance metric, which directly reflects the defining structural and retrieval model of vector databases. See Oracle AI Vector Search overview and OCI Generative AI Agents vector search documentation.

QUESTION NO: 8

Given the following code:

chain = prompt | llm

Which statement is true about LangChain Expression Language (LCEL)?

A. LCEL is a programming language used to write documentation for LangChain.

B. LCEL is a legacy method for creating chains in LangChain.

C. LCEL is a declarative and preferred way to compose chains together.

D. LCEL is an older Python library for building Large Language Models.

ANSWER: C

Explanation:

LCEL is a declarative and preferred way to compose chains together. The expression `prompt | llm` uses LCEL's pipe operator to connect compatible LangChain components into a runnable sequence: the prompt produces formatted input and the model consumes that input to generate a response. Rather than requiring developers to manually write orchestration code for each invocation step, LCEL provides a consistent composition model for prompts, models, output parsers, retrievers, and custom runnable components.

LCEL is designed around the Runnable interface, which gives composed chains standardized operations such as `invoke`, `batch`, `stream`, and asynchronous execution. This makes a chain expressed with the pipe syntax concise while retaining useful production capabilities, including streaming, parallel execution where applicable, tracing, and integration with LangChain tooling. The syntax is therefore not merely shorthand: it represents a declarative definition of how data flows between chain components. LangChain documents LCEL as its method for chaining components and describes its built-in support for optimized parallel execution, async support, and streaming. See the LangChain LCEL concepts documentation and the guide to chaining runnables in sequence.

QUESTION NO: 9

What is the purpose of Retrievers in LangChain?

A. To train Large Language Models

B. To retrieve relevant information from knowledge bases

C. To break down complex tasks into smaller steps

D. To combine multiple components into a single pipeline

ANSWER: B

Explanation:

Retrievers in LangChain are interfaces designed to return relevant documents in response to a query. Their purpose is to connect an application or language model with external knowledge sources, such as vector stores, document databases, search systems, or other indexed corpora. A retriever accepts a user question, performs the appropriate search or similarity lookup, and supplies the most relevant content as documents that can be included in the model's context.

This is a foundational part of retrieval-augmented generation (RAG). Rather than requiring a model to rely only on information encoded during training, an application can retrieve current, proprietary, or domain-specific material at request time. The retrieved documents can then ground the generated answer in the organization's knowledge base and help the application provide more relevant, evidence-based responses. LangChain defines retrievers as components that return documents given an unstructured query, while noting that they can use many different underlying storage and search implementations. See the LangChain retrievers documentation and its overview of retrieval-augmented generation.