Oracle 1Z0-184-25 Practice Questions

Shared by Rhodes on 10-11-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

In Oracle Database 23ai, which data type is used to store vector embeddings for similarity search?

Options:

A- VECTOR2
B- BLOB
C- VECTOR
D- VARCHAR2

Answer:

C

Explanation:

Oracle Database 23ai introduces the VECTOR data type (C) specifically for storing vector embeddings used in similarity search, supporting dimensions and formats (e.g., FLOAT32, INT8). VECTOR2 (A) doesn't exist. BLOB (B) can store binary data, including vectors, but lacks the semantic structure and indexing support of VECTOR. VARCHAR2 (D) is for text, not numerical arrays. VECTOR is optimized for AI vector search with native indexing (e.g., HNSW, IVF), as per Oracle's documentation.

Question 2

Question Type: MultipleChoice

Which statement best describes the capability of Oracle Data Pump for handling vector data in thecontext of vector search applications?

Options:

A- Data Pump only exports and imports vector data if the vector embeddings are stored as BLOB (Binary Large Object) data types in the database
B- Data Pump treats vector embeddings as regular text strings, which can lead to data corruption or loss of precision when transferring vector data for vector search

C- Data Pump provides native support for exporting and importing tables containing vector data types, facilitating the transfer of vector data for vector search applications
D- Because of the complexity of vector data, Data Pump requires a specialized plug-in to handle the export and import operations involving vector data types

Answer:

C

Explanation:

Oracle Data Pump in 23ai natively supports the VECTOR data type (C), allowing export and import of tables with vector columns without conversion or plug-ins. This facilitates vector search application migrations, preserving dimensional and format integrity (e.g., FLOAT32). BLOB storage (A) isn't required; VECTOR is a distinct type. Data Pump doesn't treat vectors as text (B), avoiding corruption; it handles them as structured arrays. No specialized plug-in (D) is needed; native support is built-in. Oracle's Data Pump documentation confirms seamless handling of VECTOR data.

Question 3

Question Type: MultipleChoice

What is the significance of splitting text into chunks in the process of loading data into Oracle AI Vector Search?

Options:

A- To reduce the computational burden on the embedding model
B- To facilitate parallel processing of the data during vectorization
C- To minimize token truncation as each vector embedding model has its own maximum token limit

Answer:

C

Explanation:

Splitting text into chunks (C) in Oracle AI Vector Search (e.g., via DBMS_VECTOR_CHAIN.UTL_TO_CHUNKS) ensures that each segment fits within the token

limit of embedding models (e.g., 512 tokens for BERT), preventing truncation that loses semantic content. This improves vector quality for similarity search. Reducing computational burden (A) is a secondary effect, not the primary goal. Parallel processing (B) may occur but isn't the main purpose; chunking is about model compatibility. Oracle's documentation emphasizes chunking to align with embedding model constraints.

Question 4

Question Type: MultipleChoice

What happens when you attempt to insert a vector with an incorrect number of dimensions into a VECTOR column with a defined number of dimensions?

Options:

A- The database truncates the vector to fit the defined dimensions
B- The database pads the vector with zeros to match the defined dimensions
C- The database ignores the defined dimensions and inserts the vector as is
D- The insert operation fails, and an error message is thrown

Answer:

D

Explanation:

In Oracle Database 23ai, a VECTOR column with a defined dimension count (e.g., VECTOR(4, FLOAT32)) enforces strict dimensional integrity to ensure consistency for similarity search and indexing. Attempting to insert a vector with a mismatched number of dimensions---say, TO_VECTOR('[1.2, 3.4, 5.6]') (3D) into a VECTOR(4)---results in the insert operation failing with an error (D), such as ORA-13199: 'vector dimension mismatch.' This rigidity protects downstream AI operations; a 3D vector in a 4D column would misalign with indexed data (e.g., HNSW graphs), breaking similarity calculations like cosine distance, which require uniform dimensionality.

Option A (truncation) is tempting but incorrect; Oracle doesn't silently truncate [1.2, 3.4, 5.6] to [1.2, 3.4]---this would discard data arbitrarily, risking semantic loss (e.g., a truncated sentence embedding losing meaning). Option B (padding with zeros) seems plausible---e.g., [1.2, 3.4, 5.6] becoming [1.2, 3.4, 5.6, 0]---but Oracle avoids implicit padding to prevent unintended semantic shifts (zero-padding could alter distances). Option C (ignoring dimensions) only applies to undefined VECTOR columns (e.g., VECTOR without size), not fixed ones; here, the constraint is enforced. The failure (D) forces developers to align data explicitly (e.g., regenerate embeddings), ensuring reliability---a strict but necessary design choice in Oracle's AI

framework. In practice, this error prompts debugging upstream data pipelines, avoiding silent failures that could plague production AI systems.

Question 5

Question Type: MultipleChoice

Which function should you use to determine the storage format of a vector?

Options:

A- VECTOR_DIMENSION_FORMAT
B- VECTOR_CHUNKS
C- VECTOR_NORM
D- VECTOR_EMBEDDING

Answer:

A

Question 6

Question Type: MultipleChoice

When generating vector embeddings outside the database, What is the most suitable option for storing the embeddings for later use?

Options:

A- In a CSV file
B- In a binary FVEC file with the relational data in a CSV file
C- In the database as BLOB (Binary Large Object) data
D- In a dedicated vector database

Answer:

D

Explanation:

When vector embeddings are generated outside the database, the storage choice must balance efficiency, scalability, and usability for similarity search. A CSV file (A) is simple and human-readable but inefficient for large-scale vector operations due to text parsing overhead and lack of indexing support. A binary FVEC file (B) offers a compact format for vectors, reducing storage size and improving read performance, but separating relational data into a CSV complicates integration and querying, making it suboptimal for unified workflows. Storing embeddings as BLOBs in a relational database (C) integrates well with structured data and supports SQL access, but it lacks the specialized indexing (e.g., HNSW, IVF) and query optimizations that dedicated vector databases provide. A dedicated vector database (D), such as Milvus or Pinecone (or Oracle 23ai's vector capabilities if internal), is purpose-built for high-dimensional vectors, offering efficient storage, advanced indexing, and fast approximate nearest neighbor (ANN) searches. For external generation scenarios, where embeddings are not immediately integrated into Oracle 23ai, a dedicated vector database is the most suitable due to its performance and scalability advantages. Oracle's AI Vector Search documentation indirectly supports this by emphasizing optimized vector storage for search efficiency, though it focuses on in-database solutions.

Question 7

Question Type: MultipleChoice

How is the security interaction between Autonomous Database and OCI Generative AI managed Given Select AI?

Options:

A- By encrypting all communication between the Autonomous Database and OCI Generative AI using TLS/SSL protocols
B- By utilizing Resource Principals, which grant the Autonomous Database instance access to OCI Generative AI without exposing sensitive credentials
C- By establishing a secure VPN tunnel between the Autonomous Database and OCI Generative AI service
D- By requiring users to manually enter their OCI API keys each time they execute a natural language query

Answer:

B

Explanation:

In Oracle Database 23ai's Select AI, security between the Autonomous Database and OCI Generative AI is managed using Resource Principals (B). This mechanism allows the database instance to authenticate itself to OCI services without hardcoding credentials, enhancing security by avoiding exposure of sensitive keys. TLS/SSL encryption (A) is used for data-in-transit security, but it's a complementary layer, not the primary management method. A VPN tunnel (C) is unnecessary within OCI's secure infrastructure and not specified for Select AI. Manual API key entry (D) is impractical and insecure for automated database interactions. Oracle's documentation on Select AI highlights Resource Principals as the secure, scalable authentication method.

Question 8

Question Type: MultipleChoice

What is a key advantage of using GoldenGate 23ai for managing and distributing vector data for AI applications?

Options:

A- Real-time vector data updates across locations
B- Automatic translation of vector embeddings between formats
C- Specialized vector embedding compression
D- Built-in version control for vector data

Answer:

A

Explanation:

Oracle GoldenGate 23ai is a real-time data replication and integration tool, extended in 23ai to handle the VECTOR data type for AI applications. Its key advantage (A) is enabling real-time updates of vector data across distributed locations---e.g., replicating VECTOR columns from a primary database in New York to a secondary in London with sub-second latency. This ensures AI models (e.g., for similarity search or RAG) access the latest embeddings as source data (e.g., documents) changes, critical for dynamic environments like customer support systems where new queries demand current context. Imagine a VECTOR column storing embeddings of support tickets; GoldenGate keeps these synchronized across regions, minimizing staleness that could degrade AI responses.

Option B (automatic translation) is fictional; GoldenGate doesn't convert vector formats (e.g., FLOAT32 to INT8)---that's a model or application task. Option C (compression) isn't a GoldenGate feature; compression might occur at the storage layer, but GoldenGate focuses on

replication fidelity, not size reduction. Option D (version control) misaligns with GoldenGate's purpose; it ensures data consistency, not historical versioning like Git. Real-time replication (A) stands out, as Oracle's documentation emphasizes GoldenGate's role in keeping vector-driven AI applications globally consistent, a game-changer for distributed AI deployments where latency or inconsistency could disrupt user trust. Without this, static exports (e.g., Data Pump) would lag, undermining real-time AI use cases.

To Get Premium Files for 1Z0-184-25 Visit

https://www.p2pexams.com/products/1z0-184-25

For More Free Questions Visit

https://www.p2pexams.com/oracle/pdf/1z0-184-25

