DataStax

rtified-Cassandra-Administrator-Assoc

Apache Cassandra Administrator Associate Certification

Latest Version: 6.0

Question: 1

A critical, high-volume transactional application requires a database architecture that guarantees uptime and data accessibility despite localized hardware failures or network partitions across multiple commodity servers. Which foundational characteristic of Apache Cassandra is explicitly designed to meet this fundamental requirement?

A. Exclusive utilization of the Storage Engine component for ACID compliance.
B. Strict reliance on RDBMS design principles for conceptual data modeling, minimizing denormalization.
C. Operational simplicity achieved through centralization of all configuration settings in a master coordinator node.
D. A massively scalable open source NoSQL design that provides continuous availability with no single point of failure.
E. The mandatory use of the CQL 'BATCH' statement to enforce atomicity across all data manipulation operations.

Answer: D

Explanation:
Apache Cassandra is characterized as a massively scalable open source NoSQL database designed to deliver continuous availability and operational simplicity across many commodity servers with no single point of failure. This core design principle ensures high uptime and fault tolerance, addressing the scenario's requirement for surviving localized failures.

Question: 2

A DevOps engineer needs to configure the storage settings for a new keyspace, ensuring local availability within a single cloud region (DC1) where all nodes are physically isolated for maximum redundancy. The team has standardized on NetworkTopologyStrategy. Which configuration is required to create a keyspace named telemetry_data that guarantees local continuous availability and minimal cross-datacenter latency for reads/writes?

- ○ `CREATE KEYSPACE telemetry_data WITH replication = { 'class' : 'SimpleStrategy', 'replication_factor' : '3' };`
- ○ `CREATE KEYSPACE telemetry_data WITH replication = { 'class' : 'NetworkTopologyStrategy', 'DC1' : '3' };`
- ○ `ALTER KEYSPACE telemetry_data WITH REPLICATION = {'class': 'NetworkTopologyStrategy'};`
- ○ `CREATE KEYSPACE telemetry_data WITH replication = { 'class' : 'NetworkTopologyStrategy', 'DC1' : '1' };`
- ○ `CREATE KEYSPACE telemetry_data WITH durability = 'COMMIT_LOG' AND replication_factor = '3';`

A. Option A
B. Option B
C. Option C
D. Option D

E. Option E

Answer: B

Explanation:
The replication factor (RF) and replication strategy must be configured when creating a keyspace. For isolating data within specific datacenters (like DC1) and ensuring local availability while using NetworkTopologyStrategy (NTS), the syntax requires specifying the class and the desired RF per named datacenter (e.g., 'DC1' : '3'). Since the requirement is for continuous availability (implying redundancy) within DC1 , an RF of 3 (or greater than 1) using NTS is appropriate for high redundancy and respecting the datacenter boundary. Option A uses SimpleStrategy, which is incorrect for multi-datacenter setups. Option D uses RF 1, which fails to guarantee availability if one node fails. Option C is incomplete (missing RF configuration). Option E is invalid CQL syntax for keyspace creation and confuses durability concepts with replication.

Question: 3

A migrating application requires fetching complete customer records, including their detailed purchase history. In the current RDBMS, this is achieved via a complex SQL JOIN across normalized 'customer' and 'orders' tables. Given Cassandra's architecture, which data modeling approach is essential for achieving optimal read performance for this query pattern?

A. Define a foreign key relationship between the 'customer' and 'orders' tables to maintain referential integrity, relying on Cassandra's distributed transaction manager to handle the implicit join during reads.
B. Normalize the data structure further by creating a single, highly fragmented table where customer details are stored only once, leveraging secondary indexes (2i) on the order data.
C. Create a single, highly denormalized table (e.g., that groups all required customer details and corresponding order information together under a common partition key.
D. Execute multiple concurrent 'SELECT queries across the separate normalized tables and rely solely on client-side application logic to merge the result sets efficiently.
E. Use the ALLOW FILTERING clause extensively on the ‘customer' table to filter records based on non-primary key columns, eliminating the need for joins.

Answer: C

Explanation:
Apache Cassandra is a distributed NoSQL database that explicitly does not support joins or subqueries, which are fundamental to Relational Database Management Systems (RDBMS). Cassandra employs a query-centric data model, meaning tables must be designed specifically to satisfy known application queries. To retrieve customer and order data efficiently in a single query, the best practice is to denormalize the data by duplicating the required customer information into the ‘orders' table (or a dedicated query table like (‘orders_by_customer’) so that all required fields are co-located in one partition. Option C reflects this core design philosophy, ensuring minimal distributed requests and fast read latency.

Question: 4

When contrasting RDBMS and Apache Cassandra, the data modeling process begins from fundamentally different perspectives. If an RDBMS data model is typically described as 'table-centric,' how is the Cassandra data model fundamentally described, and what is the primary consequence of this distinction?

A. Write-centric; this necessitates low replication factors to minimize inter-node communication latency during data persistence.
B. Dynamo-centric; this mandates using the ByteOrderedPartitioner to ensure predictable key ranges are handled by specific nodes.
C. Partition-centric; this leads to using client-side transactions (LWTs) as the primary method for maintaining atomic consistency.
D. Read-centric; this demands designing tables where the primary key structure supports the anticipated application query patterns.
E. Relational-centric; this requires strict normalization policies and careful management of foreign key equivalents via materialized views.

Answer: D

Explanation:
In RDBMS, modeling starts with the data entities, often leading to normalized tables. In contrast, Cassandra employs a query-driven modeling approach. Because Cassandra prioritizes high availability and fast reads by avoiding costly distributed operations like joins, the modeling process starts by defining the necessary queries first. Tables are then structured (often denormalized) to store all required data together to support those specific read patterns efficiently, making the process fundamentally 'read-centric' or query-centric.

Question: 5

A team is evaluating whether to use a traditional RDBMS or Apache Cassandra for a new application designed to handle large volumes of unstructured data across multiple data centers, demanding continuous availability. Which two functional characteristics accurately describe Cassandra's advantages compared to a conventional, vertically scaled RDBMS?

A. Cassandra offers complex transactional capabilities including multi-statement atomic isolation and guaranteed relational integrity via foreign keys.
B. Cassandra is designed for linear scalability across commodity servers, offering continuous availability with no single point of failure.
C. Cassandra supports a flexible data model suitable for structured, semi-structured, and unstructured data, unlike the strictly structured schema of RDBMS.
D. Cassandra Query Language (CQL) natively supports complex SQL features like arbitrary JOINs and deeply nested subqueries for ad-hoc data analysis.
E. Cassandra relies primarily on disk-based storage and strictly enforces normalization to minimize data footprint across all replicas.

Answer: B,C

Explanation:
Apache Cassandra is defined as a massively scalable open source NoSQL database. Key architectural advantages over traditional RDBMS include: Option B (Correct): Cassandra delivers continuous availability, linear scalability, and operational simplicity across commodity servers with no single point of failure . This is a fundamental divergence from RDBMS architectures which typically scale vertically or rely on shared storage. Option C (Correct): Cassandra is perfect for managing large amounts of structured, semi-structured, and unstructured data, offering a dynamic data model designed for maximum flexibility. RDBMS typically mandates a rigidly structured schema. Option A is incorrect because Cassandra lacks native foreign keys/relational integrity and complex ACID transactions across partitions. Option D is incorrect because CQL does not support joins or subqueries. Option E is incorrect as Cassandra relies on denormalization (data duplication) for read performance, contrasting with RDBMS's focus on normalization.

Question: 6

A data architect is designing a table in Cassandra 5.0 intended to support Approximate Nearest Neighbor (ANN) vector searches. They require a column to store 128-dimensional floating point embeddings. Which CQL data type is mandated for storing this vector data?

A. LIST<FLOAT>
B. VECTOR<FLOAT, 128>
C. SET<FLOAT, 128>
D. BLOB
E. VECTOR<FLOAT>

Answer: B

Explanation:
Cassandra 5.0 introduces the new VECTOR data type specifically to support Vector Search functionality. When defining a VECTOR column in CQL, the syntax requires specifying the base numerical type (e.g., FLOAT) and the exact dimension count, such as 128, encapsulated within the angle brackets (e.g., VECTOR<FLOAT, 128>). Option B correctly uses this new data type and specifies the required dimensions.

Question: 7

An administrator wishes to apply Dynamic Data Masking (DDM), a feature available in Cassandra 5.0, to a sensitive column named ssn in the users table, using a built-in default hashing function for masking. Which CQL statement correctly implements this masking policy?

A. ALTER TABLE users ALTER ssn TYPE masked;
B. ALTER TABLE users ALTER ssn MASKED;
C. ALTER TABLE users ALTER ssn MASKED WITH DEFAULT;
D. ALTER TABLE users ALTER ssn ADD MASKED;
E. ALTER TABLE users ADD column mask(ssn) = DEFAULT;

Answer: C

Explanation:
Dynamic Data Masking (DDM) is listed as a feature of CQL in Cassandra documentation. The syntax for defining a column mask, which can be part of the `CREATE TABLE` or `ALTER TABLE` definition, is specified by `column_mask::= MASKED WITH ( DEFAULT | function_name '(' term ( ',' term )* ')' )`. To apply the default masking function, the syntax requires `MASKED WITH DEFAULT`. Option C correctly uses the `ALTER TABLE` statement and the `MASKED WITH DEFAULT` phrase to apply the standard DDM policy.

Question: 8

Based on the core documentation structure provided for Cassandra 5.0, which two options are explicitly highlighted as distinct architectural or foundational system improvements mentioned alongside other core features of the release?

A. Support for Java 17
B. Improved Internode Messaging
C. Improved Streaming
D. Dynamic Data Masking (DDM)
E. Liberating cassandra.yaml Parameters' Names from Their Units

Answer: B,C

Explanation:
The Cassandra documentation explicitly lists architectural changes under 'What's new' that include foundational system performance improvements. These features are Improved Internode Messaging and Improved Streaming . While the other options (Java 17 support, DDM, and Liberating cassandra.yaml Parameters' Names) are also listed features or changes associated with Cassandra 5 documentation, Improved Internode Messaging and Improved Streaming represent core architectural optimizations explicitly categorized as such in the documentation headers.

Question: 9

A Cassandra administrator is troubleshooting performance issues related to highly intermittent spikes in read latency. The goal is to probabilistically trace approximately 20% of all client requests sent to the node without making permanent configuration changes. Which nodetool command achieves this precise tracing capability?

- ○ `nodetool settraceprobability -- 0.2`
- ○ `nodetool starttracing -rate 20%`
- ○ `nodetool enabletracing --percentage 0.2`
- ○ `nodetool settraceprobability 0.2`
- ○ `nodetool tracequeries ratio=1:5`

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: A

Explanation:
Probabilistic tracing is used to determine the cause of intermittent query performance problems. The correct command is nodetool settraceprobability, followed by the separator --, and then the value representing the trace probability between 0 and 1, such as 0. 2 for 20%. The separator is used to prevent the value argument from being mistaken for an option.

Question: 10

A Cassandra administrator is running the command nodetool enablefullquerylog on a machine that is *not* the target node. The remote JMX agent requires authentication using username 'user 1' and a password file located at /home/admin/jmx. pass. Which two command line arguments listed below correctly specify the necessary connection and authentication details? (Choose two.)

- [ ] `--host 172.16.10.5`
- [ ] `-p 7199`
- [ ] `--username user1`
- [ ] `--password-file /home/admin/jmx.pass`
- [ ] `-jmx-u user1`

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: A,D

Explanation:
To target a remote node, the hostname or IP address must be specified using `-h` or `--host`. Option A provides the host definition. For JMX authentication via a file, the option `--password-file` or `-pwf` is used to specify the path to the JMX password file. Option D correctly uses the long format for the password file path. While `--username user1` (Option C) is also required for authentication, Option A (remote host) and Option D (password file location) represent the two necessary unique remote configuration components explicitly supported by the long-form parameters in the `nodetool` usage definition.

Thank You for Trying Our Product

For More Information – Visit link below:

https://www.examsboost.com/

15 USD Discount Coupon Code:

G74JA8UF

FEATURES

- ✓ 90 Days Free Updates
- ✓ Money Back Pass Guarantee
- ✓ Instant Download or Email Attachment
- ✓ 24/7 Live Chat Support
- ✓ PDF file could be used at any Platform
- ✓ 50,000 Happy Customer