DP-420^{Q&As}

Designing and Implementing Cloud-Native Applications Using Microsoft Azure Cosmos DB

Pass Microsoft DP-420 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/dp-420.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

⚙ **Instant Download** After Purchase

⚙ **100% Money Back** Guarantee

⚙ **365 Days** Free Update

⚙ **800,000+** Satisfied Customers

QUESTION 1

You have an Azure Cosmos DB for NoSQL account named account1 that has a single read-write region and one additional read region.

Account1 uses the strong default consistency level.

You have an application that uses the eventual consistency level when submitting requests to account1.

How will writes from the application be handled?

A. Writes will use the strong consistency level.

B. Azure Cosmos DB will reject writes from the application.

C. The write order is not guaranteed during replication.

D. Writes will use the eventual consistency level.

Correct Answer: A

This is because the write concern is mapped to the default consistency level configured on your Azure Cosmos DB account, which is strong in this case. Strong consistency ensures that every write operation is synchronously committed to every region associated with your Azure Cosmos DB account. The eventual consistency level that the application uses only applies to the read operations. Eventual consistency offers higher availability and better performance, but it does not guarantee the order or latency of the reads.

QUESTION 2

You have a container in an Azure Cosmos DB Core (SQL) API account. The container stores telemetry data from IoT devices. The container uses telemetryId as the partition key and has a throughput of 1,000 request units per second (RU/s).

Approximately 5,000 IoT devices submit data every five minutes by using the same telemetryId value.

You have an application that performs analytics on the data and frequently reads telemetry data for a single IoT device to perform trend analysis.

The following is a sample of a document in the container.

```
{
    "id" : "9ccf1906-2a30-4dc0-9644-2185f5dcbbd7",
    "deviceId" : "bba6fe24-6d97-4935-8d58-36baa4b8a0e1",
    "telemetryId" : "9d7816e6-f401-42ba-ad05-0e03de35c0b8",
    "date" : "2019-05-03",
    "time" : "13:05",
    "temp" : "21"
}
```

You need to reduce the amount of request units (RUs) consumed by the analytics application. What should you do?

A. Decrease the offerThroughputvalue for the container.

B. Increase the offerThroughputvalue for the container.

C. Move the data to a new container that has a partition key of deviceId.

D. Move the data to a new container that uses a partition key of date.

Correct Answer: C

The partition key is what will determine how data is routed in the various partitions by Cosmos DB and needs to make sense in the context of your specific scenario. The IoT Device ID is generally the "natural" partition key for IoT applications.

Reference: https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/iot-using-cosmos-db

QUESTION 3

You plan to store order data in Azure Cosmos DB for NoSQL account. The data contains information about orders and their associated items.

You need to develop a model that supports order read operations. The solution must minimize the number or requests.

A. Create a single database that contains one container. Store orders and order items in separate documents in the container.

B. Create a single database that contains one container. Create a separate document for each order and embed the order items into the order documents.

C. Create a database for orders and a database for order items.

D. Create a single database that contains a container for order and a container for order items.

Correct Answer: B

Azure Cosmos DB is a multi-model database that supports various data models, such as documents, key-value, graph, and column-family3. The core content- model of Cosmos DB\\'s database engine is based on atom-record-sequence (ARS), which allows it to store and query different types of data in a flexible and efficient way3. To develop a model that supports order read operations and minimizes the number of requests, you should consider the following factors:

1.

 The size and shape of your data

2.

 The frequency and complexity of your queries

3.

 The latency and throughput requirements of your application

4.

The trade-offs between storage efficiency and query performance

Based on these factors, one possible model that you could implement is B. Create a single database that contains one container. Create a separate document for each order and embed the order items into the order documents.

This model has the following advantages:

1.

 It stores orders and order items as self-contained documents that can be easily retrieved by order ID1.

2.

 It avoids storing redundant data or creating additional containers for order items1.

3.

 It allows you to view the order history of a customer with simple queries1.

4.

 It leverages the benefits of embedding data, such as reducing the number of requests, improving query performance, and simplifying data consistency2.

This model also has some limitations, such as: It may not be suitable for some order items that have data that is greater than 2 KB, as it could exceed the maximum document size limit of 2 MB2. It may not be optimal for scenarios where order items need to be queried independently from orders or aggregated by other criteria2. It may not support transactions across multiple orders or customers, as transactions are scoped to a single logical partition2. Depending on your specific use case and requirements, you may need to adjust this model or choose a different one. For example, you could use a hybrid data model that combines embedding and referencing data2, or you could use a graph data model that expresses entities and relationships as vertices and edges.

QUESTION 4

You have an Azure Cosmos DB for NoSQL account.

The change feed is enabled on a container named invoice.

You create an Azure function that has a trigger on the change feed.

What is received by the Azure function?

A. all the properties of the updated items

B. only the partition key and the changed properties of the updated items

C. all the properties of the original items and the updated items

D. only the changed properties and the system-defined properties of the updated items

Correct Answer: A

According to the Azure Cosmos DB documentation12, the change feed is a persistent record of changes to a container in the order they occur. The change feed outputs the sorted list of documents that were changed in the order in which

they were modified.

The Azure function that has a trigger on the change feed receives all the properties of the updated items2. The change feed does not include the original items or only the changed properties. The change feed also includes some system-

defined properties such as _ts (the last modified timestamp) and _lsn (the logical sequence number)3.

Therefore, the correct answer is "all the properties of the updated items"

QUESTION 5

HOTSPOT

You have an Azure Cosmos DB Core (SQL) API account named account1.

You have the Azure virtual networks and subnets shown in the following table.

Subnet	Network	IP address range	Virtual machine
subnet1	vnet1	10.0.0.0/24	VM1
subnet2	vnet1	10.0.1.0/24	VM2
subnet3	vnet2	10.1.0.0/24	VM3

The vnet1 and vnet2 networks are connected by using a virtual network peer.

The Firewall and virtual network settings for account1 are configured as shown in the exhibit.

Allow access from
○ All networks   ● Selected networks
Configure network security for your Azure Cosmos DB account. Learn more.

Virtual networks
Secure your Azure Cosmos DB account with virtual networks.   + Add existing virtual network   +Add new virtual network

Virtual Network	Subnet	Address range	Endpoint Status
∨ vnet1	1	10.0.0.0/16	
	vnet1.subnet1	10.0.1.0/24	✓ Enabled

Firewall
Add IP ranges to allow access from the internet or your on-premises networks.   +Add my current IP   ⓘ

IP(Single IPv4 or CIDR range)

Exceptions
☐ Accept connections from within public Azure datacenters ⓘ
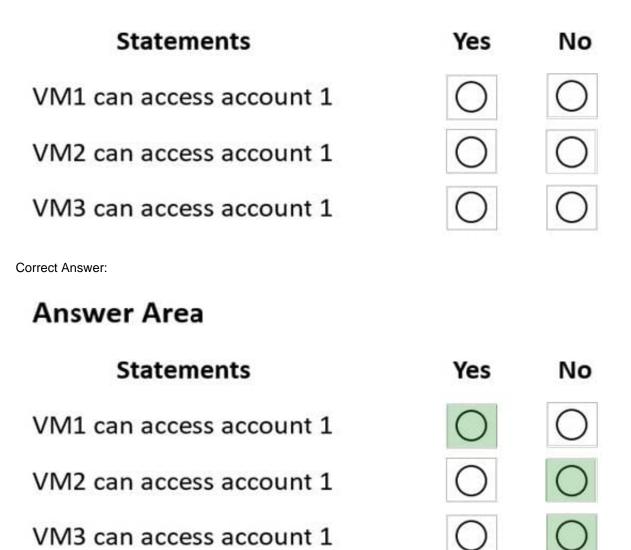☑ Allow access from Azure Portal ⓘ

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

Statements	Yes	No
VM1 can access account 1	○	○
VM2 can access account 1	○	○
VM3 can access account 1	○	○

Correct Answer:

Answer Area

Statements	Yes	No
VM1 can access account 1	●	○
VM2 can access account 1	○	●
VM3 can access account 1	○	●

Box 1: Yes

VM1 is on vnet1.subnet1 which has the Endpoint Status enabled.

Box 2: No

Only virtual network and their subnets added to Azure Cosmos account have access. Their peered VNets cannot access the account until the subnets within peered virtual networks are added to the account.

Box 3: No

Only virtual network and their subnets added to Azure Cosmos account have access.

Reference:

https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-configure-vnet-service-endpoint

<u>DP-420 PDF Dumps</u>   <u>DP-420 Study Guide</u>   <u>DP-420 Exam Questions</u>