

➤ **Vendor: Microsoft**

➤ **Exam Code: DP-420**

➤ **Exam Name: Designing and Implementing Cloud-Native Applications Using Microsoft Azure Cosmos DB**

➤ **New Updated Questions from [Braindump2go](#) (Updated in [July/2023](#))**

[Visit Braindump2go and Download Full Version DP-420 Exam Dumps](#)

QUESTION 52

You are working on an app that will save the device metrics produced every minute by various IoT devices. You decide to collect the data for two entities; the devices and the device metrics produced by each device. You were about to create two containers for these identified entities but suddenly your data engineer suggests placing both entities in a single container, not in two different containers. What would you do?

- A. Create a document with the deviceid property and other device data, and add a property called 'type' having the value 'device'. Also, create another document for each metrics data collected using devicemetricsid property.
- B. Create a document with the deviceid property and other device data. After that embed each metrics collection into the document with the devicemetricsid property and all the metrics data.
- C. Create a document with the deviceid property and other device data, and add a property called "type" with the value 'device'. Create another document for each metrics data using the devicemetricsid and deviceid properties and add a property called "type" with the value devicemetrics.
- D. None of these.

Answer: C

Explanation:

If you create two different types of documents including the property 'deviceid' for both entities, it will ease referencing both entities inside the container.

QUESTION 53

The below diagram illustrates the configuration settings for a container in an Azure Cosmos DB Core (SQL) API account.

Which of the following statements rightly describes the container's configuration?

- A. All the items will be deleted after 1 hour.
- B. All the items will be deleted after 1 month.
- C. Items will be expired only if they have a time to live value.
- D. Items stored in containers will get retained always, regardless of the time to live value of the items.

Answer: C

Explanation:

Time to live can be set on a container or an item within the container.

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/time-to-live>

QUESTION 54

[DP-420 Exam Dumps](#) [DP-420 Exam Questions](#) [DP-420 PDF Dumps](#) [DP-420 VCE Dumps](#)

<https://www.braindump2go.com/dp-420.html>

While working in .NET SDK v3, you need to enable multi-region writes in your app that uses Azure Cosmos DB. WestUS2 is the region where your application would be deployed and Cosmos DB is replicated. Which of the following attributes would you set to WestUS2 to achieve the goal?

- A. ApplicationRegion
- B. SetCurrentLocation
- C. setPreferredLocations
- D. connection_policy.PreferredLocations

Answer: A

Explanation:

In .NET SDK v3, if you want to enable multi-region writes in your application, set ApplicationRegion to the region where the application is being deployed and Cosmos DB is replicated.

<https://docs.microsoft.com/bs-latn-ba/azure/cosmos-db/sql/how-to-multi-master>

QUESTION 55

While chairing a team session, you are telling the team members about the Azure Cosmos DB Emulator. Which of the following statement is not true about Azure Cosmos DB Emulator?

- A. Azure Cosmos DB Emulator offers an emulated environment that would run on the local developer workstation.
- B. The emulator does support only a single fixed account and a familiar primary key. You can even regenerate the key while utilizing the Azure Cosmos DB Emulator.
- C. The emulator doesn't offer multi-region replication.
- D. The emulator doesn't offer various Azure Cosmos DB consistency levels as offered by the cloud service.

Answer: B

Explanation:

<https://docs.microsoft.com/en-us/azure/cosmos-db/local-emulator>

QUESTION 56

You need to configure the consistency levels on a per-request basis.

Which of the following C# class would you use in the .NET SDK for Azure Cosmos DB SQL API?

- A. CosmosClientOptions
- B. CosmosConfigOptions
- C. ItemRequestOptions
- D. Container

Answer: C

Explanation:

The ItemRequestOptions class contains various session token and consistency level configuration properties on a per-request basis.

QUESTION 57

Which of the following function in Spark SQL separates the array's elements into multiple rows with positions and utilizes the column names 'pos' for position and 'col' for elements of the array?

- A. explode()
- B. posexplode()
- C. preexplode()
- D. Separate()

Answer: B

Explanation:

posexplode() is a function in Spark SQL that separates the array's elements into multiple rows with positions and utilizes the column names pos for position and col for elements of the array.

<https://docs.microsoft.com/en-us/learn/modules/query-azure-cosmos-db-with-apache-spark-for-azure-synapse-analytics/5-perform-complex-queries>

QUESTION 58

One of the team members has recently joined the team and wants to manually adjust the time for which the items will remain in the cache. He asks you for help about which property from the ItemRequestOptions class should he configure. What would you suggest?

- A. ConsistencyLevel
- B. SessionToken
- C. MaxIntegratedCacheStaleness
- D. None of these

Answer: C

Explanation:

MaxIntegratedCacheStaleness indicates the maximum acceptable staleness for the cached queries and point reads, irrespective of the selected consistency. This property is set to configure a TimeSpan that will be used to limit how long items will remain in the cache.

<https://docs.microsoft.com/en-us/azure/cosmos-db/integrated-cache>

QUESTION 59

You work in the Intel company that has its clients across 4 different countries, with hundreds of thousands of clients for countries 1 and 2, and a few thousand clients for countries 3 and 4. Requests for every country total approximately 50,000 RU/s every hour of the day. The application team of the company suggests using countryId as the partition key for this container.

Which of the following statements is true?

- A. This partition key will cause fan out when filtering by countryId
- B. This partition key could cause storage hot partitions
- C. This partition key will prevent throughput hot partitions
- D. All the above

Answer: B

Explanation:

In the given scenario, countries 1 and 2 have hundreds of thousands of clients. Storage hot partitions are likely for countries 1 or 2 since they would have to bulk of the data stored.

Option A is incorrect. The suggested partition will prevent fan out as we will be always going through one partition per countryId.

Option B is correct. Storage hot partitions are likely for countries 1 or 2 since they would have the bulk of the data stored.

Option C is incorrect. There is still a possibility of throughput hot partitions.

Option D is incorrect. This partition key could cause storage hot partitions.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/partitioning-overview>

QUESTION 60

Gateway mode using the dedicated gateway is one of the ways to connect to an Azure Cosmos DB account. Which of the following statement(s) is/are true about a dedicated gateway?

- A. Dedicated gateways are supported only on SQL API accounts
- B. A dedicated gateway can be provisioned in Azure Cosmos DB accounts with IP firewalls or Private Link configured
- C. A dedicated gateway can be provisioned in an Azure Cosmos DB account in a Virtual Network

[DP-420 Exam Dumps](#) [DP-420 Exam Questions](#) [DP-420 PDF Dumps](#) [DP-420 VCE Dumps](#)

<https://www.braindump2go.com/dp-420.html>

(Vnet)

- D. You can use RBAC (role-based access control) to authenticate data plane requests routed through the dedicated gateway

Answer: A

Explanation:

<https://docs.microsoft.com/en-us/azure/cosmos-db/dedicated-gateway#connect-to-azure-cosmos-db-using-direct-mode>

QUESTION 61

You need to create a .NET console app to manage data in the Azure Cosmos DB SQL API account. As a part of the procedure, you need to create a database. Which of the following method can you use to create a database?

- A. BuildDatabaseAsync
- B. CreateCosmosDatabase
- C. CreateContainerAsync
- D. CreateDatabaseAsync

Answer: D

Explanation:

A database is the logical container of items that are partitioned across containers. Either the CreateDatabaseAsync or CreateDatabaseIfNotExistsAsync method of the CosmosClient class can be used to create a database.

Option A is incorrect. BuildDatabaseAsync is not a valid method to create the database.

Option B is incorrect. CreateCosmosDatabase won't help in creating the database.

Option C is incorrect. CreateContainerAsync method is used to create a container, not a database.

Option D is correct. Either the CreateDatabaseAsync or CreateDatabaseIfNotExistsAsync method of the CosmosClient class can be used to create a database.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/sql-api-get-started>

QUESTION 62

You are part of a development team and your team creates a set of aggregate metadata items that are needed to be changed anytime, once you successfully update or create an item within your container. Which of the following server-side programming constructs would you use for this task?

- A. User-defined function
- B. System-defined function
- C. Pre-trigger
- D. Post-trigger

Answer: D

Explanation:

A post-trigger runs its logic after the item has been successfully updated or created. At this point, you can update the aggregate metadata items.

Option A is incorrect. A user-defined function is used only within the context of a SQL query.

Option B is incorrect. A system-defined function can not be used in this scenario.

Option C is incorrect. A pre-trigger will run its logic too early before the item gets successfully created or updated.

Option D is correct. A post-trigger runs its logic after the item has been successfully updated or created.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-write-stored-procedures-triggers-udfs>

QUESTION 63

Which of the following is the most optimal method of managing referential integrity between different containers in Azure Cosmos DB?

- A. Creating an Azure Cosmos DB Function that will periodically search for changes done to your containers and replicate those changes to the referenced containers

[DP-420 Exam Dumps](#) [DP-420 Exam Questions](#) [DP-420 PDF Dumps](#) [DP-420 VCE Dumps](#)

<https://www.braindump2go.com/dp-420.html>

- B. Using an Azure Function trigger for Azure Cosmos DB to leverage on the change feed processor to update the referenced containers
- C. When any changes are created by your app to one container, also ensure it duplicates the changes on the referenced container
- D. All are equally efficient

Answer: B

Explanation:

The change feed APIs help in keeping the track of all the Insert and Updates on a container, creating a trigger depending upon the change feed, will allow the users to send those changes to other containers that also require those changes.

Option A is incorrect. Azure Cosmos DB functions don't modify data and their scope is limited only within the container they are defined in.

Option B is correct. Using an Azure Function trigger for Azure Cosmos DB to leverage the change feed processor to update the referenced containers is the most optimal way to manage referential integrity between different containers in Azure Cosmos DB.

Option C is incorrect. While it can be possible to add this logic to your app, adding the code to the application won't add complexity to your environment, but also it won't guarantee that any changes carried out outside of the app would be replicated to the other containers.

Option D is incorrect. Using an Azure Function trigger for Azure Cosmos DB to leverage the change feed processor to update the referenced containers is the most optimal method.

Reference:

<https://docs.microsoft.com/en-us/learn/modules/advanced-modeling-patterns-azure-cosmos-db/4-manage-referential-integrity-with-change-feed>

QUESTION 64

App1 is an application that reads the data stored in the Azure Cosmos DB Core (SQL) API account. The application runs the same read queries every minute. Eventual is the default consistency level set for the account.

You identify that instead of using the cache, the request units are consumed by every query.

You verify that both the metrics IntegratedCacheItemHitRate and IntegratedCacheQueryHitRate are having values of 0. You also verify the provisioning of the dedicated gateway cluster and its use in the connection string.

You need to ensure that the application App1 uses the Azure Cosmos DB integrated cache. Which of the following do you need to configure?

- A. The consistency level of the requests from the application App1
- B. The indexing policy of the Azure Cosmos DB container
- C. The default consistency level of the Azure Cosmos DB account
- D. The connectivity mode of the application App1 CosmosClient

Answer: D

Explanation:

As the integrated cache is specific to the Azure Cosmos DB account and needs significant memory and CPU, it needs a dedicated gateway node. Therefore, you need to configure the connectivity mode of the application CosmosClient. Connect to Azure Cosmos DB with the help of gateway mode.

Option A is incorrect. You need to configure the connectivity mode of App1 CosmosClient, not the consistency level of the requests.

Option B is incorrect. There is no need to configure the indexing policy.

Option C is incorrect. Eventual is the default consistency level set for the account. Therefore, configuring the default consistency level of the Azure Cosmos DB account is not the right answer.

Option D is correct. You need to configure the connectivity mode of the application CosmosClient. Connect to Azure Cosmos DB using gateway mode.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/integrated-cache-faq>

QUESTION 65

Container2 is a container in an Azure Cosmos DB Core (SQL) API account where Upserts of items happen every three seconds.

[DP-420 Exam Dumps](#) [DP-420 Exam Questions](#) [DP-420 PDF Dumps](#) [DP-420 VCE Dumps](#)

<https://www.braindump2go.com/dp-420.html>

You are having an Azure Functions application named function2 that is expected to run whenever the items are replaced or inserted in container2.

You notice that function2 runs, but not on each upsert.

You need to make sure that each upsert is processed by function2 within 1 second of the upsert.

Which of the following property should you modify in the Function.json file of function2?

- A. MaxItemsPerInvocation
- B. FeedPollDelay
- C. CheckpointInterval
- D. leaseCollectionsThroughput

Answer: B

Explanation:

Upsert operation is an operation where a row is inserted into a database table if it does not already exist, or updated if it exists. FeedPollDelay represents the delay time (in ms) between the poll of a partition for new updates upon the feed after all current modifications are drained. The default value for this is 5000 ms i.e. 5 seconds.

Therefore, if you want that each upsert is processed by function2 within 1 second of the upsert, you need to change the FeedPollDelay.

Option A is incorrect. MaxItemsPerInvocation is used to set the maximum items received for each Function call.

Option B is correct. There is a need to change the FeedPollDelay property.

Option C is incorrect. CheckpointInterval defines the interval between lease checkpoints. The Default is always after each Function call.

Option D is incorrect. leaseCollectionsThroughput is not the right property to change.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-cosmosdb-v2-trigger>

QUESTION 66

Range, spatial, and composite are three different types of indexes supported by Azure Cosmos DB.

Which of these indexes are used for geospatial objects like points, polygons, LineStrings, etc?

- A. Range Indexes
- B. Composite Indexes
- C. Spatial Indexes
- D. None of these

Answer: C

Explanation:

Spatial Indexes are used for geospatial objects. Currently, it supports Points, Polygons, MultiPolygons, and LineStrings. Spatial indexes can also be used on GeoJSON objects.

Option A is incorrect. Range indexes are not used on geospatial objects.

Option B is incorrect. Use composite indexes when you need to enhance the efficiency of queries that execute operations on several fields.

Option C is correct. Spatial indexes are utilized for geospatial objects.

Option D is incorrect. The spatial index is the right answer.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/index-overview>

QUESTION 67

The server-side latency direct and server-side latency gateway are two metrics that can be used to view the server-side latency of any operation in two different connection modes. Which of the following API(s) in Azure Cosmos DB supports both of these metrics?

- A. SQL
- B. MongoDB
- C. Cassandra
- D. Gremlin

[DP-420 Exam Dumps](#) [DP-420 Exam Questions](#) [DP-420 PDF Dumps](#) [DP-420 VCE Dumps](#)

<https://www.braindump2go.com/dp-420.html>

E. Table

Answer: AE

Explanation:

The below table demonstrates which API supports server-side latency metrics (Direct vs Gateway):

Command	Purpose
cd, chdir	Changes the parent directory.
md, mkdir	Creates a directory.
del, erase	Deletes one or more files.
Move	Moves one or multiple files.

Option A is correct. SQL API supports both Server-Side Latency Direct as well as Server-Side Latency Gateway.

Option B is incorrect. MongoDB supports only Server-Side Latency Gateway.

Option C is incorrect. Cassandra supports only Server-Side Latency Gateway.

Option D is incorrect. Gremlin supports does not support Server-Side Latency Direct.

Option E is correct. Table supports both Server-Side Latency Direct as well as Server-Side Latency Gateway.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/monitor-server-side-latency>

QUESTION 68

You have a database named db1 in an Azure Cosmos DB Core (SQL) API account.

You are designing an application that will use db1.

In db1, you are creating a new container named coll1 that will store online orders.

The following is a sample of a document that will be stored in coll1.

```
{
  "customerId" : "bba6fe24-6d97-4935-8d58-36baa4b8a0e1",
  "orderId" : "9d7816e6-f401-42ba-ad05-0e03de35c0b8",
  "orderDate" : "2021-09-29",
  "orderDetails" : []
}
```

The application will have the following characteristics:

- New orders will be created frequently by different customers.
- Customers will often view their past order history.

You need to select the partition key value for coll1 to support the application. The solution must minimize costs.

To what should you set the partition key?

- A. orderId
- B. customerId
- C. orderDate
- D. id

Answer: B

Explanation:

If most of your workload's requests are queries and most of your queries have an equality filter on the same property, this property can be a good partition key choice. For example, if you frequently run a query that filters on UserID, then selecting UserID as the partition key would reduce the number of cross-partition queries.

[DP-420 Exam Dumps](#) [DP-420 Exam Questions](#) [DP-420 PDF Dumps](#) [DP-420 VCE Dumps](#)

<https://www.braindump2go.com/dp-420.html>

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/partitioning-overview>

QUESTION 69

You have a container in an Azure Cosmos DB Core (SQL) API account that stores data about orders. The following is a sample of an order document.

```
{  
  "orderId" : "d4a9179b-5ead-43a3-b851-add9a71ac4b6",  
  "customerId" : "f6e39103-bdc7-4346-9cfb-45daa4b2becf",  
  "orderDate" : "2021-09-29",  
  "orderItems" : [...],  
  "total" : 12345  
}
```

Documents are up to 2 KB.

You plan to receive one million orders daily.

Customers will frequently view their past order history.

You are evaluating whether to use orderDate as the partition key.

What are two effects of using orderDate as the partition key? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. You will exceed the maximum number of partition key values
- B. You will exceed the maximum storage per partition
- C. There will always be a hot partition
- D. Queries will run cross-partition

Answer: CD

Explanation:

Not A: There is no limit to the total number of physical partitions in your container. As your provisioned throughput or data size grows, Azure Cosmos DB will automatically create new physical partitions by splitting existing ones

Not B: 1 million a day x 2KB -> 2 GB of data/day while maximum storage across all items per (logical) partition: 20 GB

QUESTION 70

You have a container in an Azure Cosmos DB Core (SQL) API account. The container stores data about families. Data about parents, children, and pets are stored as separate documents.

Each document contains the address of each family. Members of the same family share the same partition key named familyId.

You need to update the address for each member of the same family that share the same address. The solution must meet the following requirements:

- Be atomic, consistent, isolated, and durable (ACID).
- Provide the lowest latency.

What should you do?

- A. Update the document of each family member separately by using a patch operation
- B. Update the document of each family member separately and set the consistency level to strong
- C. Update the document of each family member by using a transactional batch operation

Answer: C

Explanation:

Each transaction provides ACID (Atomicity, Consistency, Isolation, Durability) property guarantees.

Transactional batch operations offer reduced latency on equivalent operations.

Note: Transactional batch describes a group of point operations that need to either succeed or fail together with the same partition key in a container. In the .NET

SDK, the TransactionalBatch class is used to define this batch of operations. If all operations succeed in the order they are described within the transactional batch operation, the transaction will be committed. However, if any operation

[DP-420 Exam Dumps](#) [DP-420 Exam Questions](#) [DP-420 PDF Dumps](#) [DP-420 VCE Dumps](#)

<https://www.braindump2go.com/dp-420.html>

fails, the entire transaction is rolled back.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/transactional-batch>

QUESTION 71

You are designing an Azure Cosmos DB Core (SQL) API solution to store data from IoT devices. Writes from the devices will occur every second.

The following is a sample of the data.

```
{
  "id" : "03c1ca5a-db18-4231-908f-09a9bc7a7c3e",
  "deviceManufacturer" : "Contoso, Ltd",
  "deviceId" : "f460df85-799f-4d58-b051-67561b4993c6",
  "timestamp" : "2021-09-19T13:47:45",
  "sensor1Value" : true,
  "sensor2Value" : "75",
  "sensor3Value" : "4554",
  "sensor4Value" : "454",
  "sensor5Value" : "42128"
}
```

You need to select a partition key that meets the following requirements for writes:

- Minimizes the partition skew
- Avoids capacity limits
- Avoids hot partitions

What should you do?

- A. Create a new synthetic key that contains deviceId and timestamp
- B. Use timestamp as the partition key
- C. Use deviceManufacturer as the partition key
- D. Use sensor1Value as the partition key

Answer: A

Explanation:

Concatenate multiple properties of an item.

You can form a partition key by concatenating multiple property values into a single artificial partitionKey property.

These keys are referred to as synthetic keys.

For example, consider the following example document:

```
{
  "deviceId": "abc-123",
  "date": 2018
}
```

For the previous document, one option is to set /deviceId or /date as the partition key. Use this option, if you want to partition your container based on either device

ID or date. Another option is to concatenate these two values into a synthetic partitionKey property that's used as the partition key.

```
{
  "deviceId": "abc-123",
  "date": 2018,
  "partitionKey": "abc-123-2018"
}
```

Incorrect:

Not B: But the problem is that when the application writes new data, the writes will always be directed to the same partition, based on whatever day it is. This results in what's called a hot partition, where we have a bottleneck that's going to quickly consume a great deal more of the reserved throughput you've provisioned for the container.

[DP-420 Exam Dumps](#) [DP-420 Exam Questions](#) [DP-420 PDF Dumps](#) [DP-420 VCE Dumps](#)

<https://www.braindump2go.com/dp-420.html>

Specifically, Cosmos DB evenly distributes your provisioned throughput across all the physical partitions in the container.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/synthetic-partition-keys>

<https://docs.microsoft.com/en-us/azure/cosmos-db/concepts-limits>

QUESTION 72

You need to create a data store for a directory of small and medium-sized businesses (SMBs). The data store must meet the following requirements:

- Store companies and the users employed by them. Each company will have less than 1,000 users.
- Some users have data that is greater than 2 KB.
- Associate each user to only one company.
- Provide the ability to browse by company.
- Provide the ability to browse the users by company.
- Whenever a company or user profile is selected, show a details page for the company and all the related users.
- Be optimized for reading data.

Which design should you implement to optimize the data store for reading data?

- A. In a directory container, create a document for each company and a document for each user. Use the company ID as the partition key.
- B. Create a user container that uses the user ID as the partition key and a company container that uses the company ID as the partition key. Add the company ID to each user document.
- C. In a user container, create a document for each user. Embed the company into each user document. Use the user ID as the partition key.
- D. In a company container, create a document for each company. Embed the users into company documents. Use the company ID as the partition key.

Answer: D

Explanation:

All employees within a company would nicely fit within a single document (document size 2 MB).

Note: An Azure Cosmos container is the unit of scalability both for provisioned throughput and storage. A container is horizontally partitioned and then replicated across multiple regions. The items that you add to the container are automatically grouped into logical partitions, which are distributed across physical partitions, based on the partition key.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/set-throughput>