Exam Questions DP-420

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

https://www.2passeasy.com/dumps/DP-420/

NEW QUESTION 1
- (Exam Topic 1)
You need to select the partition key for con-iot1. The solution must meet the IoT telemetry requirements. What should you select?

A. the timestamp
B. the humidity
C. the temperature
D. the device ID

Answer: D

Explanation:
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.
Scenario: The iotdb database will contain two containers named con-iot1 and con-iot2. Ensure that Azure Cosmos DB costs for IoT-related processing are predictable. Reference:
https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/iot-using-cosmos-db

NEW QUESTION 2
- (Exam Topic 1)
You configure multi-region writes for account1.
You need to ensure that App1 supports the new configuration for account1. The solution must meet the business requirements and the product catalog requirements.
What should you do?

A. Set the default consistency level of accountl to bounded staleness.
B. Create a private endpoint connection.
C. Modify the connection policy of App1.
D. Increase the number of request units per second (RU/s) allocated to the con-product and con-productVendor containers.

Answer: D

Explanation:
App1 queries the con-product and con-productVendor containers.
Note: Request unit is a performance currency abstracting the system resources such as CPU, IOPS, and memory that are required to perform the database operations supported by Azure Cosmos DB.
Scenario:
Develop an app named App1 that will run from all locations and query the data in account1.
Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.
Whenever there are multiple solutions for a requirement, select the solution that provides the best performance, as long as there are no additional costs associated.
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/consistency-levels

NEW QUESTION 3
- (Exam Topic 2)
You have an Azure Cosmos DB Core (SQL) API account named account1.
In account1, you run the following query in a container that contains 100GB of data. SELECT *
FROM c
WHERE LOWER(c.categoryid) = "hockey"
You view the following metrics while performing the query.

```
Retrieved Document Count                   :            45,654
Retrieved Document Size                     :     543,765,234 bytes
Output Document Count                        :                12
Output Document Size                          :           451 bytes
Index Utilization                                :           0.00 %
Total Query Execution Time                  :     2,400.34 milliseconds
   Query Preparation Times
      Query Compilation Time                :          0.09 milliseconds
      Logical Plan Build Time               :          0.04 milliseconds
      Physical Plan Build Time              :          0.03 milliseconds
      Query Optimization Time               :          0.01 milliseconds
   Index Lookup Time                          :          0.00 milliseconds
   Document Load Time                         :      3,167.26 milliseconds
   Runtime Execution Times
      Query Engine Times                     :        299.16 milliseconds
      System Function Execution Time        :         79.34 milliseconds
      User-defined Function Execution Time  :          0.00 milliseconds
   Document Write Time                        :          0.01 milliseconds
Client Side Metrics
   Retry Count                                  :                0
   Request Charge                               :      3,898.95 RUs
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

Answer Area

Statements	Yes	No
The query performs a cross-partition query	○	○
The query uses an index	○	○
Recreating the container with the partition key set to /categoryId will improve the performance of the query	○	○

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Box 1: No
Each physical partition should have its own index, but since no index is used, the query is not cross-partition.
Box 2: No
Index utilization is 0% and Index Look up time is also zero.
Box 3: Yes
A partition key index will be created, and the query will perform across the partitions. Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-query-container

NEW QUESTION 4
- (Exam Topic 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 offerThroughput value for the container.
B. Increase the offerThroughput value 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.

Answer: C

Explanation:
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

NEW QUESTION 5
- (Exam Topic 2)
You have the following query.
SELECT * FROM
WHERE c.sensor = "TEMP1"
AND c.value < 22
AND c.timestamp >= 1619146031231
You need to recommend a composite index strategy that will minimize the request units (RUs) consumed by the query.
What should you recommend?

A. a composite index for (sensor ASC, value ASC) and a composite index for (sensor ASC, timestamp ASC)
B. a composite index for (sensor ASC, value ASC, timestamp ASC) and a composite index for (sensor DESC, value DESC, timestamp DESC)
C. a composite index for (value ASC, sensor ASC) and a composite index for (timestamp ASC, sensor ASC)
D. a composite index for (sensor ASC, value ASC, timestamp ASC)

Answer: A

Explanation:
If a query has a filter with two or more properties, adding a composite index will improve performance. Consider the following query:
SELECT * FROM c WHERE c.name = "Tim" and c.age > 18
In the absence of a composite index on (name ASC, and age ASC), we will utilize a range index for this query. We can improve the efficiency of this query by creating a composite index for name and age.
Queries with multiple equality filters and a maximum of one range filter (such as >,<, <=, >=, !=) will utilize the composite index.

Reference:
https://azure.microsoft.com/en-us/blog/three-ways-to-leverage-composite-indexes-in-azure-cosmos-db/

NEW QUESTION 6
- (Exam Topic 2)
You are developing an application that will use an Azure Cosmos DB Core (SQL) API account as a data source.
You need to create a report that displays the top five most ordered fruits as shown in the following table.

Name	Type	Orders
apple	fruit	1,000
orange	fruit	600
banana	fruit, exotic	400
plum	fruit.	300
mango	fruit, exotic	200

A collection that contains aggregated data already exists. The following is a sample document:
{
"name": "apple",
"type": ["fruit", "exotic"], "orders": 10000
}
Which two queries can you use to retrieve data for the report? Each correct answer presents a complete solution.
NOTE: Each correct selection is worth one point.
A)

```
SELECT TOP  i.name, i.types, i.orders
FROM items i
WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit')
ORDER BY i.orders,i.types
```

B)

```
SELECT TOP  i.name, i.types, i.orders
FROM items i
WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit')
ORDER BY i.orders DESC
```

C)

```
SELECT TOP  i.name, i.types, i.orders
FROM items i
WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit')
ORDER BY i.types DESC
```

D)

```
SELECT TOP  i.name, i.types, i.orders
FROM items i
WHERE ARRAY_CONTAINS(i.types, {name: 'fruit'})
ORDER BY i.orders DESC
```

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

Answer: BD

Explanation:
ARRAY_CONTAINS returns a Boolean indicating whether the array contains the specified value. You can check for a partial or full match of an object by using a boolean expression within the command.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/sql/sql-query-array-contains

NEW QUESTION 7
- (Exam Topic 2)
You have an Azure Cosmos DB Core (SQL) API account that uses a custom conflict resolution policy. The account has a registered merge procedure that throws a runtime exception.
The runtime exception prevents conflicts from being resolved. You need to use an Azure function to resolve the conflicts. What should you use?

A. a function that pulls items from the conflicts feed and is triggered by a timer trigger
B. a function that receives items pushed from the change feed and is triggered by an Azure Cosmos DB trigger
C. a function that pulls items from the change feed and is triggered by a timer trigger
D. a function that receives items pushed from the conflicts feed and is triggered by an Azure Cosmos DB trigger

Answer: D

Explanation:
The Azure Cosmos DB Trigger uses the Azure Cosmos DB Change Feed to listen for inserts and updates across partitions. The change feed publishes inserts and updates, not deletions.
Reference: https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-cosmosdb

NEW QUESTION 8
- (Exam Topic 2)
You have an Azure Cosmos DB Core (SQL) API account named account1 that has the disableKeyBasedMetadataWriteAccess property enabled.
You are developing an app named App1 that will be used by a user named DevUser1 to create containers in account1. DevUser1 has a non-privileged user account in the Azure Active Directory (Azure AD) tenant.
You need to ensure that DevUser1 can use App1 to create containers in account1. What should you do? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

Answer Area

Grant permissions to create containers by using:
- Account keys
- Resource tokens
- Role-based access control (RBAC)

Create containers by using the:
- Azure AD Graph API
- Azure Resource Manager API
- SQL (Core) API

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Box 1: Resource tokens
Resource tokens provide access to the application resources within a database. Resource tokens:
Provide access to specific containers, partition keys, documents, attachments, stored procedures, triggers, and UDFs.
Box 2: Azure Resource Manager API
You can use Azure Resource Manager to help deploy and manage your Azure Cosmos DB accounts, databases, and containers.
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/secure-access-to-data https://docs.microsoft.com/en-us/rest/api/resources/

NEW QUESTION 9
- (Exam Topic 2)
You are designing an Azure Cosmos DB Core (SQL) API solution to store data from IoT devices. Writes from the devices will be 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. Use timestamp as the partition key.
B. Create a new synthetic key that contains deviceId and sensor1Value.
C. Create a new synthetic key that contains deviceId and deviceManufacturer.
D. Create a new synthetic key that contains deviceId and a random number.

Answer: D

Explanation:
Use a partition key with a random suffix. Distribute the workload more evenly is to append a random number at the end of the partition key value. When you distribute items in this way, you can perform parallel write operations across partitions.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/sql/synthetic-partition-keys

NEW QUESTION 10
- (Exam Topic 2)
You need to implement a trigger in Azure Cosmos DB Core (SQL) API that will run before an item is inserted into a container.
Which two actions should you perform to ensure that the trigger runs? Each correct answer presents part of the solution.
NOTE: Each correct selection is worth one point.

A. Append pre to the name of the JavaScript function trigger.
B. For each create request, set the access condition in RequestOptions.
C. Register the trigger as a pre-trigger.
D. For each create request, set the consistency level to session in RequestOptions.
E. For each create request, set the trigger name in RequestOptions.

Answer: C

Explanation:
C: When triggers are registered, you can specify the operations that it can run with.
F: When executing, pre-triggers are passed in the RequestOptions object by specifying PreTriggerInclude and then passing the name of the trigger in a List object.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-use-stored-procedures-triggers-udfs

NEW QUESTION 10
- (Exam Topic 2)
You have an application named App1 that reads the data in an Azure Cosmos DB Core (SQL) API account. App1 runs the same read queries every minute. The default consistency level for the account is set to eventual.
You discover that every query consumes request units (RUs) instead of using the cache.
You verify the IntegratedCacheiteItemHitRate metric and the IntegratedCacheQueryHitRate metric. Both metrics have values of 0.
You verify that the dedicated gateway cluster is provisioned and used in the connection string. You need to ensure that App1 uses the Azure Cosmos DB integrated cache.
What should you configure?

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

Answer: C

Explanation:
Because the integrated cache is specific to your Azure Cosmos DB account and requires significant CPU and memory, it requires a dedicated gateway node.
Connect to Azure Cosmos DB using gateway mode.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/integrated-cache-faq

NEW QUESTION 12
- (Exam Topic 2)
You have an Azure Cosmos DB Core (SQL) API account that is used by 10 web apps.
You need to analyze the data stored in the account by using Apache Spark to create machine learning models. The solution must NOT affect the performance of the web apps.
Which two actions should you perform? Each correct answer presents part of the solution. NOTE: Each correct selection is worth one point.

A. In an Apache Spark pool in Azure Synapse, create a table that uses cosmos.olap as the data source.
B. Create a private endpoint connection to the account.
C. In an Azure Synapse Analytics serverless SQL pool, create a view that uses OPENROWSET and the CosmosDB provider.
D. Enable Azure Synapse Link for the account and Analytical store on the container.
E. In an Apache Spark pool in Azure Synapse, create a table that uses cosmos.oltp as the data source.

Answer: AD

Explanation:
Reference:
https://github.com/microsoft/MCW-Cosmos-DB-Real-Time-Advanced-Analytics/blob/main/Hands-on%20lab/H

NEW QUESTION 14
- (Exam Topic 2)
You have an Azure Cosmos DB Core (SQL) account that has a single write region in West Europe. You run the following Azure CLI script.

```
az cosmosdb update -n $accountName -g $resourceGroupName \
    --locations regionName='West Europe' failoverPriority=0 isZoneRedundant-False \
    --locations regionName='North Europe' failoverPriority=1 isZoneRedundant-False

az cosmosdb failover-priority-change -n $accountName -g $resourceGroupName \
    --failover-policies 'North Europe=0' 'West Europe=1'
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

Answer Area

Statements	Yes	No
After running the script, there will be an instance of Azure Cosmos DB in North Europe that is writable	○	○
After running the script, the Azure Cosmos DB instance in West Europe will be writable	○	○
The cost of the Azure Cosmos DB account is unaffected by running the script	○	○

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Box 1: Yes
The Automatic failover option allows Azure Cosmos DB to failover to the region with the highest failover priority with no user action should a region become unavailable.
Box 2: No
West Europe is used for failover. Only North Europe is writable. To Configure multi-region set UseMultipleWriteLocations to true.
Box 3: Yes
Provisioned throughput with single write region costs $0.008/hour per 100 RU/s and provisioned throughput with multiple writable regions costs $0.016/per hour per 100 RU/s.
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-multi-master https://docs.microsoft.com/en-us/azure/cosmos-db/optimize-cost-regions

NEW QUESTION 18
- (Exam Topic 2)
You have a database named telemetry in an Azure Cosmos DB Core (SQL) API account that stores IoT data. The database contains two containers named readings and devices.
Documents in readings have the following structure.
id
deviceid
timestamp
ownerid
measures (array)
- type
- value
- metricid
Documents in devices have the following structure.
id
deviceid
owner
- ownerid
- emailaddress
- name brand model
For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

Answer Area

Statements	Yes	No
To return for all devices owned by a specific `emailaddress`, multiple queries must be performed	○	○
To return `deviceid`, `ownerid`, `timestamp`, and `value` for a specific `metricid`, a join must be performed	○	○
To return `deviceid`, `ownerid`, `emailaddress`, and `model`, a join must be performed	○	○

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Box 1: Yes
Need to join readings and devices.
Box 2: No
Only readings is required. All required fields are in readings.
Box 3: No
Only devices is required. All required fields are in devices.

NEW QUESTION 19
- (Exam Topic 2)
You are creating a database in an Azure Cosmos DB Core (SQL) API account. The database will be used by an application that will provide users with the ability to share online posts. Users will also be able to submit comments on other users' posts.
You need to store the data shown in the following table.

Type	Description
Users	Information about a user who will use the application
Posts	Text of up to 1,000 characters that a user will share with other users
Comments	Text of up to 280 characters that users will submit as a comment on a post
Interests	Information about a user's interests

The application has the following characteristics: Users can submit an unlimited number of posts.

The average number of posts submitted by a user will be more than 1,000. Posts can have an unlimited number of comments from different users.
The average number of comments per post will be 100, but many posts will exceed 1,000 comments. Users will be limited to having a maximum of 20 interests.
For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

Answer Area

Statements	Yes	No
If you embed the posts data into the users data instead of creating a separate document for each post, you will increase the write operation costs for new posts	⭕	⭕
If you embed the comments data into the posts data instead of creating a separate document for each comment you will increase the write operation costs for new comments	⭕	⭕
If you embed the interests data into the users data instead of creating a separate document for each interest, you will increase the read operation costs for displaying the users and their associated interests	⭕	⭕

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Box 1: Yes
Non-relational data increases write costs, but can decrease read costs.
Box 2: Yes
Non-relational data increases write costs, but can decrease read costs.
Box 3: No
Non-relational data increases write costs, but can decrease read costs.

NEW QUESTION 20
- (Exam Topic 2)
You have a container named container1 in an Azure Cosmos DB Core (SQL) API account. Upserts of items in container1 occur every three seconds.
You have an Azure Functions app named function1 that is supposed to run whenever items are inserted or replaced in container1.
You discover that function1 runs, but not on every upsert.
You need to ensure that function1 processes each upsert within one second of the upsert. Which property should you change in the Function.json file of function1?

A. checkpointInterval
B. leaseCollectionsThroughput
C. maxItemsPerInvocation
D. feedPollDelay

Answer: D

Explanation:
With an upsert operation we can either insert or update an existing record at the same time.
FeedPollDelay: The time (in milliseconds) for the delay between polling a partition for new changes on the feed, after all current changes are drained. Default is 5,000 milliseconds, or 5 seconds.
Reference: https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-cosmosdb-v2-trigger

NEW QUESTION 24
- (Exam Topic 2)
Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.
You have an Azure Cosmos DB Core (SQL) API account named account 1 that uses autoscale throughput. You need to run an Azure function when the normalized request units per second for a container in account1 exceeds a specific value.
Solution: You configure an Azure Monitor alert to trigger the function. Does this meet the goal?

A. Yes
B. No

Answer: A

Explanation:
You can set up alerts from the Azure Cosmos DB pane or the Azure Monitor service in the Azure portal. Note: Alerts are used to set up recurring tests to monitor the availability and responsiveness of your Azure
Cosmos DB resources. Alerts can send you a notification in the form of an email, or execute an Azure Function when one of your metrics reaches the threshold or if a specific event is logged in the activity log.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/create-alerts

NEW QUESTION 25
......

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual DP-420 Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the DP-420 Product From:

https://www.2passeasy.com/dumps/DP-420/

Money Back Guarantee

DP-420 Practice Exam Features:

* DP-420 Questions and Answers Updated Frequently

* DP-420 Practice Questions Verified by Expert Senior Certified Staff

* DP-420 Most Realistic Questions that Guarantee you a Pass on Your FirstTry

* DP-420 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year