Microsoft.DP-420.v2026-09-07.q139

Exam Code:	DP-420
Exam Name:	Designing and Implementing Cloud-Native Applications Using Microsoft Azure Cosmos DB
Certification Provider:	Microsoft
Free Question Number:	139
Version:	v2026-09-07
# of views:	106
# of Questions views:	1443
https://www.freecram.net/torrent/Microsoft.DP-420.v2026-09-07.q139.html	

NEW QUESTION: 1
Hotspot Question
You have an Azure Cosmos DB for NoSQL account that contains a container named container1.
You plan to use container1 as a key-value store that will perform point reads on the item ID and partition key.
You need to define an indexing policy for container1. The solution must meet the following requirements:
- Provide the ability to set DefaultTimeToLive for container1.
- Minimize implementation time.
- Minimize costs.
How should you complete the JSON definition of the indexing policy? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
{
    "indexingMode": [ ▼ ]
                    "consistent",
                    "lazy",
                    "none",
    "automatic": false,

    "includedPaths": [ ▼ ]
                    [],
                    [{"path": "/\"_etag\"/?"}],
                    [{"path": "/\"_ts\"/?"}],
    "excludedPaths":[{"path": "/*"}]

}
```

Microsoft
freecram.net

Answer:

Answer Area

```
{
    "indexingMode": [ ▼ ]
                    "consistent",
                    "lazy",
                    "none",
    "automatic": false,
    "includedPaths": [ ▼ ]
                    [],
                    [{"path": "/\"_etag\"/?"}],
                    [{"path": "/\"_ts\"/?"}],
    "excludedPaths":[{"path": "/*"}]

}
```

Microsoft
freecram.net

NEW QUESTION: 2

Hotspot Question

You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account named account1 and a Log Analytics workspace named Workspace1. Workspace1 stores the logs of account1.

You need to identify which operations used the most request units per second (RU/s) during the last 24 hours.

How should you complete the query? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
[ AzureActivity | AzureDiagnostics | AzureMetrics ]
| where ResourceProvider=="MICROSOFT.DOCUMENTDB" and Category== [ "ControlPlaneRequests" | "DataPlaneRequests" | "PartitionKeyStatistics" ]

| where TimeGenerated >= ago(1d)

| summarize max(responseLength_s), max(requestLength_s), max(requestCharge_s), count = count() by OperationName,
requestResourceType_s, userAgent_s, collectionRid_s
```

Answer:

Answer Area

```
[ AzureActivity | AzureDiagnostics | AzureMetrics ]
| where ResourceProvider=="MICROSOFT.DOCUMENTDB" and Category== [ "ControlPlaneRequests" | "DataPlaneRequests" | "PartitionKeyStatistics" ]

| where TimeGenerated >= ago(1d)

| summarize max(responseLength_s), max(requestLength_s), max(requestCharge_s), count = count() by OperationName,
requestResourceType_s, userAgent_s, collectionRid_s
```

freecram.net Microsoft

NEW QUESTION: 3

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 a database in an Azure Cosmos DB for NoSQL account that is configured for multi- region writes.
You need to use the Azure Cosmos DB SDK to implement the conflict resolution policy for a container. The solution must ensure that any conflicts are sent to the conflicts feed.
Solution: You set ConflictResolutionMode to LastWriterWins and you use the default settings for the policy.
Does this meet the goal?
A. Yes
B. No
Answer: (SHOW ANSWER)
The question states "The solution must ensure that any conflicts are sent to the conflicts feed." Last-writer-wins explicitly states - "Conflicts resolved with this policy do not show up in the conflict feed. "

NEW QUESTION: 4
You plan to create an Azure Cosmos DB account that will use an integrated cache. What will prevent you from using an integrated cache?
A. a user-assigned managed identity
B. Azure Private Link
C. a default consistency level of Session
D. availability zones
Answer: (SHOW ANSWER)
Integrated cache in Azure Cosmos DB cannot be used in conjunction with Azure Private Link.
Private Link provides a private endpoint for securely accessing Azure services without exposing them to the public internet, but it currently does not support integration with the Cosmos DB integrated cache.

NEW QUESTION: 5
Hotspot Question
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account. The account hosts a container named Container1.
Items are stored in Container1 in the following format.

```
{
    "id": "97841ca4-68ab-4cd9-506f-11216d30d71e",
    "productId": "USBGold",
    "name": "USB Key - Gold Color",
    "category": "USB Storage",
    "availableOptions":
    [
      {
        "sizeInGb": "4",
        "price": 2.78,
        "currentStockLevel": 143
      },
      {
        "sizeInGb": "8",
        "price": 3.75,
        "currentStockLevel": 42
      },
      {
        "sizeInGb": "16",
        "price": 4.95,
        "currentStockLevel": 177
      }
    ]
}
```


freecram.net

Microsoft

You need to query name, sizeInGb, and currentStockLevel for items that have a value of currentStockLevel that is greater than 100.

How should you complete the query? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
SELECT d.name, ao.sizeInGb, ao.currentStockLevel

FROM drives d

[ ▼ ] ao
  APPLY
  INNER JOIN
  JOIN
  OUTER JOIN

[ ▼ ] d.availableOptions
  CROSS APPLY
  IN
  JOIN
  OUTER APPLY

WHERE ao.currentStockLevel > 100
```

freecram.net

Answer:

Answer Area

```
SELECT d.name, ao.sizeInGb, ao.currentStockLevel

FROM drives d

[ ▼ ]  ao

[ ▼ ]  d.availableOptions
```

- APPLY
- INNER JOIN
- JOIN
- OUTER JOIN

- CROSS APPLY
- IN
- JOIN
- OUTER APPLY

freecram.net

```
OUTER APPLY

WHERE ao.currentStockLevel > 100
```

NEW QUESTION: 6
You have an Azure Cosmos DB for NoSQL container.
All queries to the container use the Bounded Staleness consistency level.
You discover that when many queries are executed against the container, an HTTP 429 error is returned.
What are two ways to reduce the number of HTTP 429 errors? Each correct answer presents a complete solution.
NOTE: Each correct selection is worth one point.
A. Use a query request option to change the consistency level to Strong.
B. Add a post-trigger to the container.
C. Add a pre-trigger to the container.
D. Add query retry logic to the client application.
E. Increase the maximum throughput.
Answer: (SHOW ANSWER)
[E]
Diagnose and troubleshoot "Request rate too large" (429) exceptions
A "Request rate too large" exception, also known as error code 429, indicates that your requests against Azure Cosmos DB are being rate limited.
You can increase the RU/s on the database or container using the client SDKs, Azure portal, PowerShell, CLI, or ARM template.
[B]
Strong consistency
Strong consistency offers a linearizability guarantee. Linearizability means serving requests concurrently. The reads are guaranteed to return the most recent committed version of an item. A client never sees an uncommitted or partial write. Users are always guaranteed to read the latest committed write.
Note: Azure Cosmos DB offers five well-defined levels. From strongest to weakest, the levels are:
Strong
Bounded staleness
Session
Consistent prefix
Eventual
Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/troubleshoot-request-rate-too-large
https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels

NEW QUESTION: 7
Hotspot Question
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account.

```
var taxIdPath = new ClientEncryptionIncludedPath
{
    Path = "/taxId",
    ClientEncryptionKeyId = "cek1",
    EncryptionType = EncryptionType.Deterministic.ToString(),
    EncryptionAlgorithm = DataEncryptionAlgorithm.AeadAes256CbcHmacSha256
};
var creditCardPath = new ClientEncryptionIncludedPath
{
    Path = "/creditCard",
    ClientEncryptionKeyId = "cek1",
    EncryptionType = EncryptionType.Randomized.ToString(),
    EncryptionAlgorithm = DataEncryptionAlgorithm.AeadAes256CbcHmacSha256
};
await database.DefineContainer("customers", "/customerID")
    .WithClientEncryptionPolicy()
    .WithIncludedPath(taxIdPath)
    .WithIncludedPath(creditCardPath)
    .Attach()
```

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
Queries can retrieve items from the Customers container by using the `taxId` field.	○	○
If two customers have the same credit card value, the encrypted fields will contain the same value.	○	○
The Customers container is created if the container does **NOT** exist. If the container exists, the policy of the container is updated.	○	○

Answer:

freecram.net Microsoft

Answer Area

Statements	Yes	No
Queries can retrieve items from the Customers container by using the `taxId` field.	○	○
If two customers have the same credit card value, the encrypted fields will contain the same value.	○	○
The Customers container is created if the container does **NOT** exist. If the container exists, the policy of the container is updated.	○	○

Microsoft
freecram.net

NEW QUESTION: 8
You have an Azure Cosmos DB for NoSQL database named db1 that writes to multiple Azure regions.
You need to test the performance of db1 in the secondary region.
Which command should you run first?
A. az cosmosdb location show
B. az cosmosdb failover-priority-change
C. az cosmosdb network-rule remove
D. az cosmosdb sql database merge
Answer: (SHOW ANSWER)
To perform a manual failover in Azure Cosmos DB for NoSQL, use the 'az cosmosdb failover- priority-change' Azure CLI command with the -failover-policies option, changing the priority of the desired region to 0.
Note:
az cosmosdb failover-priority-change
Changes the failover priority for the Azure Cosmos DB database account.
Incorrect:
[Not A]
The Azure CLI command to show the properties of an Azure Cosmos DB location is az cosmosdb locations show.
[Not B]
az cosmosdb failover-priority-change
Changes the failover priority for the Azure Cosmos DB database account.
[Not D]
az cosmosdb network-rule remove
Remove a virtual network rule from an existing Cosmos DB database account.
Reference:
https://learn.microsoft.com/en-us/cli/azure/cosmosdb?view=azure-cli-latest#az-cosmosdb-failover-priority-change

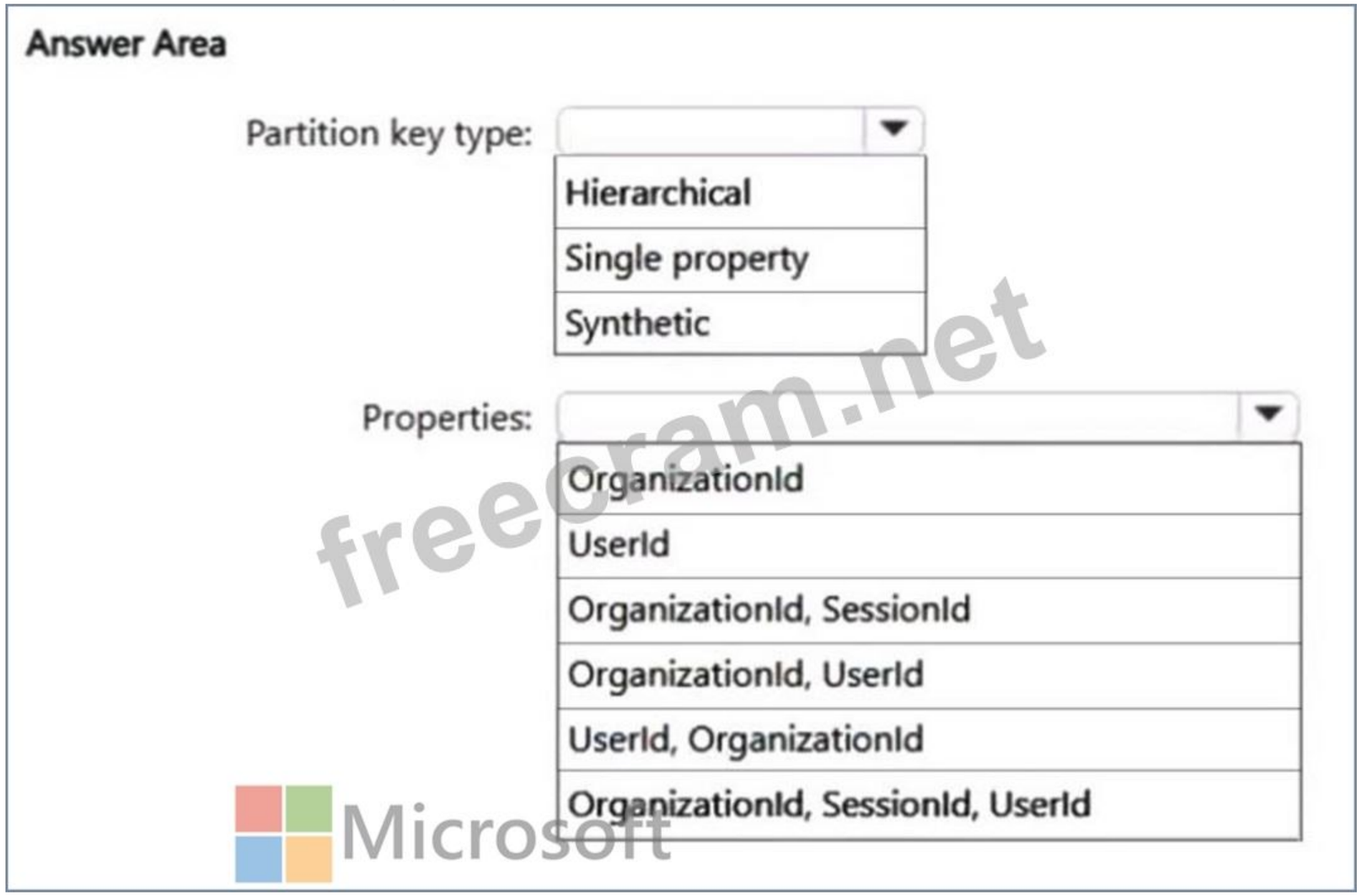
NEW QUESTION: 9
You have an Azure subscription.
You plan to implement an Azure Cosmos DB for NoSQL single-region write account named account1 that will replicate across two Azure regions.
You need to set the default consistency level for account1. The solution must meet the following requirements:
- Writes must be performed to a local majority in a single region.
- Reads must be performed from a local minority in a single region.
Which consistency level should you select?
A. Session
B. Consistent Prefix
C. Strong
D. Bounded Staleness
Answer: (SHOW ANSWER)
Based on the Azure Cosmos DB architecture for a NoSQL single-region write account with two regions, the appropriate consistency level is Bounded Staleness.
Here is how this level maps to your specific requirements:
Writes to a Local Majority: For all consistency levels except Strong, Azure Cosmos DB performs writes to a local majority (three out of four replicas) within the primary write region before acknowledging the operation.
Reads from a Local Minority: In the Bounded Staleness and Strong consistency levels, reads are performed against a local minority (two replicas in a four-replica set) to compare log sequence numbers (LSN) and ensure consistency.
Single-Region Write vs. Strong Consistency: While Strong consistency also uses local minority reads, it requires a global majority for writes (acknowledgment from all regions), which contradicts your requirement for a "local majority" write.
Regional Behavior: Bounded Staleness is specifically recommended for single-region write accounts with multiple regions where "near strong" consistency is desired; it provides strong consistency guarantees for clients reading within the same region as the write.
Incorrect:
Other levels like Session, Consistent Prefix, and Eventual perform reads from only a single replica, which does not meet your "local minority" read requirement.
Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels

NEW QUESTION: 10
Hotspot Question
You plan to create a container named Container1 in an Azure Cosmos DB for NoSQL account.
Container1 will store items that contain user sign-in session data from multiple organizations.
Each user is assigned to a single organization.
Container1 will include the following properties:
- UserId
- SessionId
- SessionStart
- OrganizationID
The following are common queries served by Container1.
SELECT SessionID,SessionStart,UserID FROM c WHERE OrganizationId = 6 AND UserId = 50143
SELECT SessionID,SessionStart,UserID FROM c WHERE OrganizationId = 6
You need to recommend a partitioning scheme for Container1 that meets the following requirements:
- Maximizes performance

- Minimizes the cost of read queries
- Minimizes the possibility of reaching the partition size limits

Which type of partition key and which partition key properties should you include in the recommendation? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.



freecram.net

Answer:

Answer Area



Partition key type:

- Hierarchical
- Single property
- Synthetic

Properties:

- OrganizationId
- UserId
- OrganizationId, SessionId
- OrganizationId, UserId
- UserId, OrganizationId
- OrganizationId, SessionId, UserId

freecram.net

NEW QUESTION: 11
You plan to create an Azure Cosmos DB for NoSQL account that will have a single write region and three read regions.
You need to set the consistency level for the account. The solution must meet the following requirements:
- In the write region, writes must replicate synchronously across at
least three replicas.
- In the read regions, reads must see writes in order for transactional batches.
- Throughput for reads and writes must be maximized.
Which consistency level should you select?
A. Bounded Staleness
B. Consistent Prefix
C. Eventual
D. Strong
Answer: (SHOW ANSWER)
NEW QUESTION: 12
Case Study 1 - Litware, inc

Overview

Litware, Inc. is a United States-based grocery retailer. Litware has a main office and a primary datacenter in Seattle. The company has 50 retail stores across the United States and an emerging online presence. Each store connects directly to the internet. Existing environment.

Cloud and Data Service Environments. Litware has an Azure subscription that contains the resources shown in the following table.

Name	Type	Description
account1	Azure Cosmos DB for NoSQL account	The account has a single read-write region and contains a database named productdb. The productdb database contains two containers named con-product and con-productVendor. The account uses the session default consistency level.
iothub1	Azure IoT hub	The IoT hub collects per-second temperature and humidity telemetry from 5,000 IoT devices in the retail stores. A single telemetry reading generates 1 KB of data.
streamanalytics1	Azure Stream Analytics job	The job processes telemetry data collected from iothub1.

freecram.net Microsoft

Each container in productdb is configured for manual throughput. The con-product container stores the company's product catalog data. Each document in con-product includes a con- productvendor value. Most queries targeting the data in con-product are in the following format.

SELECT * FROM con-product p WHERE p.con-productVendor - 'name'

Most queries targeting the data in the con-productVendor container are in the following format SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Existing environment.

Current Problems.

Litware identifies the following issues:

- Updates to product categories in the con-productVendor container do not propagate automatically to documents in the con-product container.
- Application updates in con-product frequently cause HTTP status code 429 "Too many requests".
- You discover that the 429 status code relates to excessive request unit (RU) consumption during the updates.

Requirements.

Planned Changes

Litware plans to implement a new Azure Cosmos DB for NoSQL account named account2 that will contain a database named iotdb. The iotdb database will contain two containers named con- iot1 and con-iot2.

Litware plans to make the following changes:

- Store the telemetry data in account2.
- Configure account1 to support multiple read-write regions.
- Implement referential integrity for the con-product container.
- Use Azure Functions to send notifications about product updates to different recipients.
- Develop an app named App1 that will run from all locations and query the data in account1.
- Develop an app named App2 that will run from the retail stores and query the data in account2.
- App2 must be limited to a single DNS endpoint when accessing account2.

Requirements.

Business Requirements

Litware identifies the following business requirements:

- 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.
- Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.
- Minimize the number of firewall changes in the retail stores.

Requirements.

Product Catalog Requirements

Litware identifies the following requirements for the product catalog:
- Implement a custom conflict resolution policy for the product catalog data.
- Minimize the frequency of errors during updates of the con-product container.
- Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.
- Trigger the execution of two Azure functions following every update to any document in the con- product container.

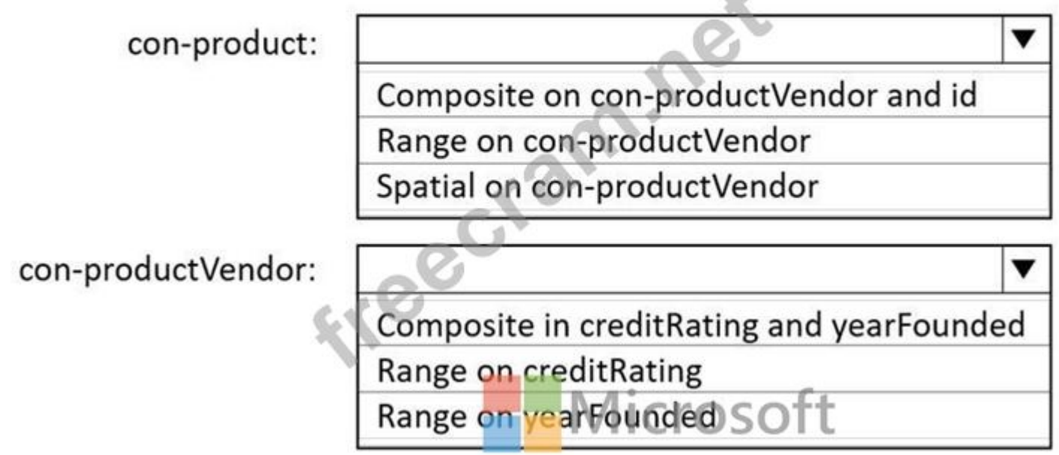
Hotspot Question

You need to recommend indexes for con-product and con-productVendor. The solution must meet the product catalog requirements and the business requirements.

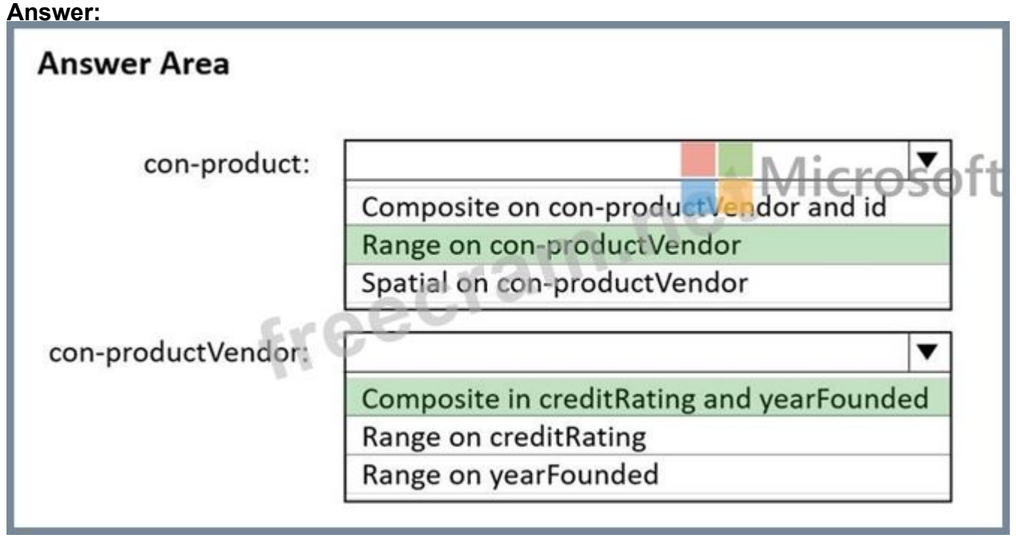
Which type of index should you recommend for each container? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area



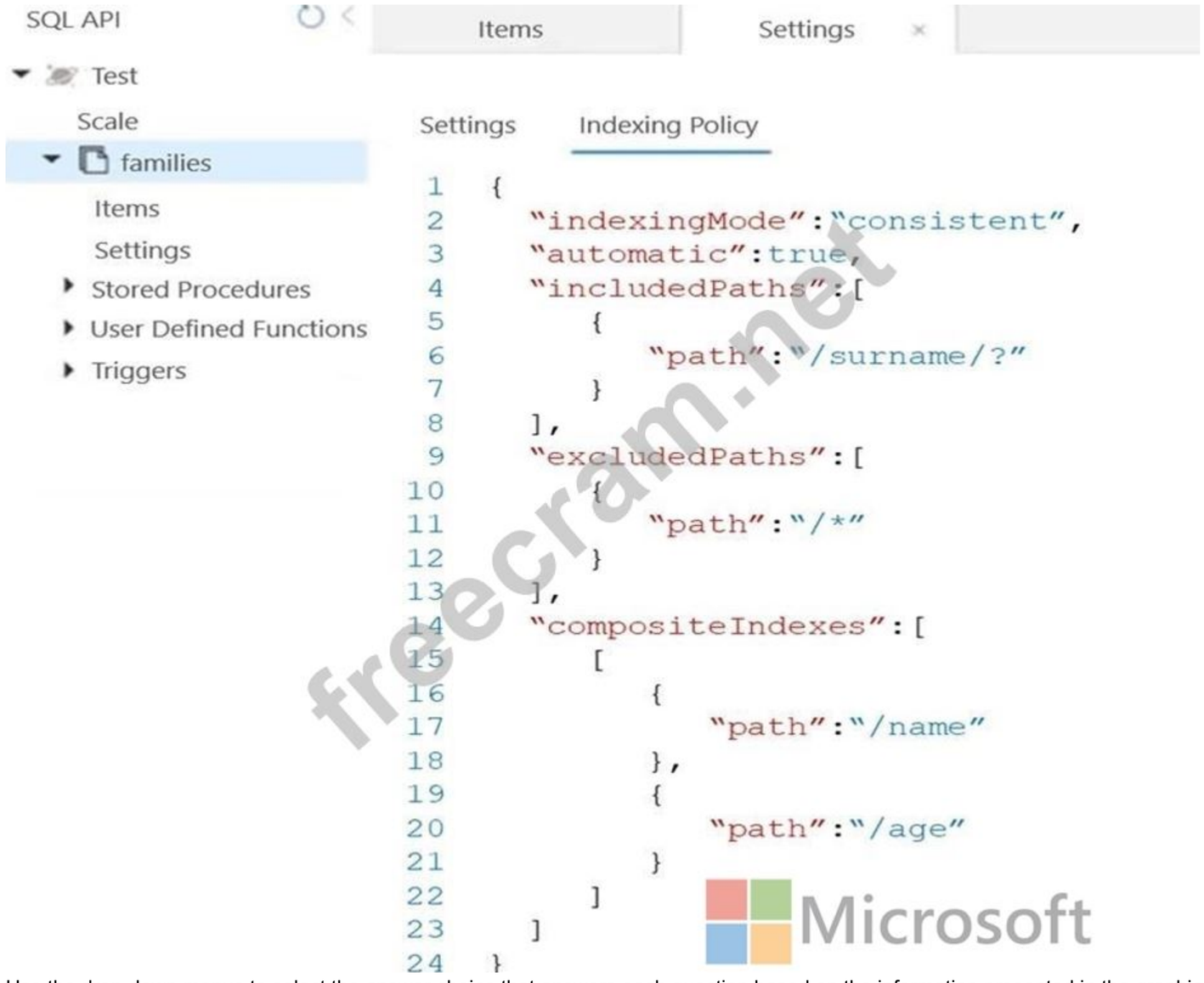
Answer:



Explanation:

Box 1: Range on con-productVendor
Range index is based on an ordered tree-like structure. The range index type is used for:
Equality queries:
SELECT * FROM container c WHERE c.property = 'value'
SELECT * FROM c WHERE c.property IN ("value1", "value2", "value3")
Note:
The con-product container stores the company's product catalog data. Each document in con- product includes a con-productvendor value. Most queries targeting the data in con-product are in the following format.
SELECT * FROM con-product p WHERE p.con-productVendor - 'name'
Box 2: Composite in creditRating and yearFounded
Composite indexes increase the efficiency when you are performing operations on multiple fields.
The composite index type is used for:
ORDER BY queries on multiple properties:
SELECT * FROM container c ORDER BY c.property1, c.property2
Note: Most queries targeting the data in the con-productVendor container are in the following format SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/index-overview

NEW QUESTION: 13
Hotspot Question
You have the indexing policy shown in the following exhibit.
\



Use the drop-down menus to select the answer choice that answers each question based on the information presented in the graphic.

NOTE: Each correct selection is worth one point.

When creating a query, which ORDER BY statement will execute successfully?

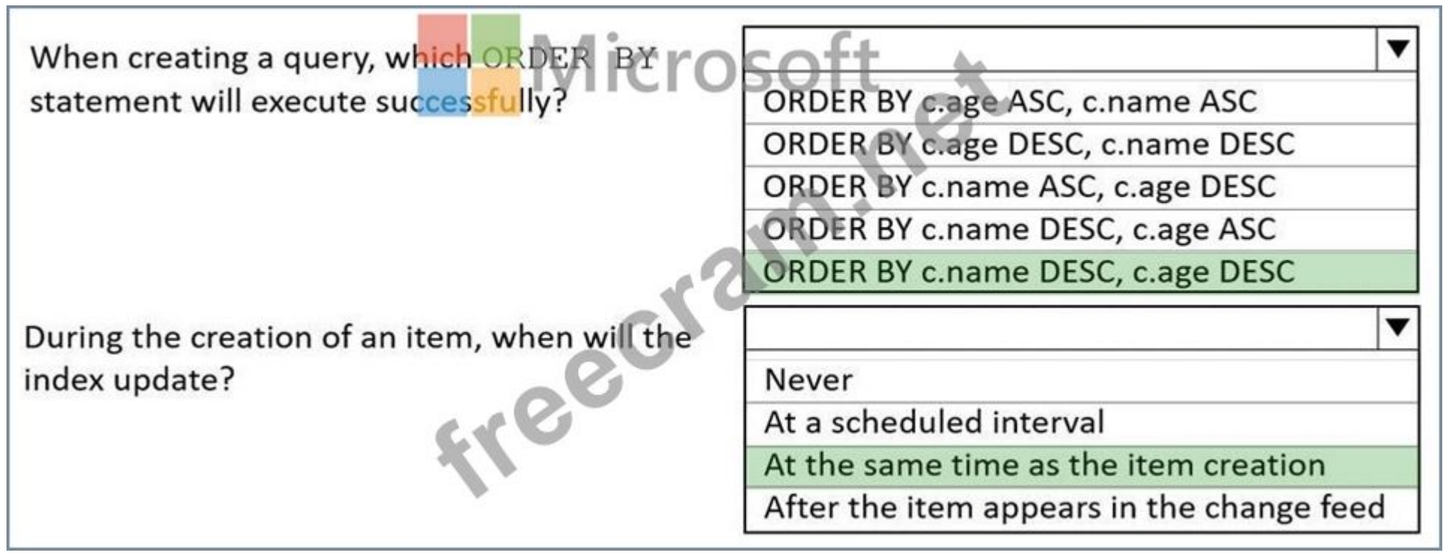
- ORDER BY c.age ASC, c.name ASC
- ORDER BY c.age DESC, c.name DESC
- ORDER BY c.name ASC, c.age DESC
- ORDER BY c.name DESC, c.age ASC
- ORDER BY c.name DESC, c.age DESC

During the creation of an item, when will the index update?

- Never
- At a scheduled interval
- At the same time as the item creation
- After the item appears in the change feed

freecram.net Microsoft

Answer:



Microsoft freecram.net

Explanation:

Box 1: ORDER BY c.name DESC, c.age DESC

Queries that have an ORDER BY clause with two or more properties require a composite index.

The following considerations are used when using composite indexes for queries with an ORDER BY clause with two or more properties:

If the composite index paths do not match the sequence of the properties in the ORDER BY clause, then the composite index can't support the query.

The order of composite index paths (ascending or descending) should also match the order in the ORDER BY clause.

The composite index also supports an ORDER BY clause with the opposite order on all paths.

Box 2: At the same time as the item creation

Azure Cosmos DB supports two indexing modes:

Consistent: The index is updated synchronously as you create, update or delete items. This means that the consistency of your read queries will be the consistency configured for the account.
None: Indexing is disabled on the container.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/index-policy

NEW QUESTION: 14
Hotspot Question
You have an Azure Cosmos DB for NoSQL container that contains the following item.



You need to update the OrderQty value to 5 by using a patch operation.
How should you complete the JSON Patch document? To answer, select the appropriate options in the answer area.



Answer:



NEW QUESTION: 15
Hotspot Question
You have an Azure Cosmos DB for NoSQL container named Contacts that is configured as shown in the following exhibit.

Container Id

contacts

Indexing

Automatic Off

All properties in your documents will be indexed by default fo
flexible and efficient queries. Learn more

Partition key

/companyID

Unique keys

/familyName,/givenNames

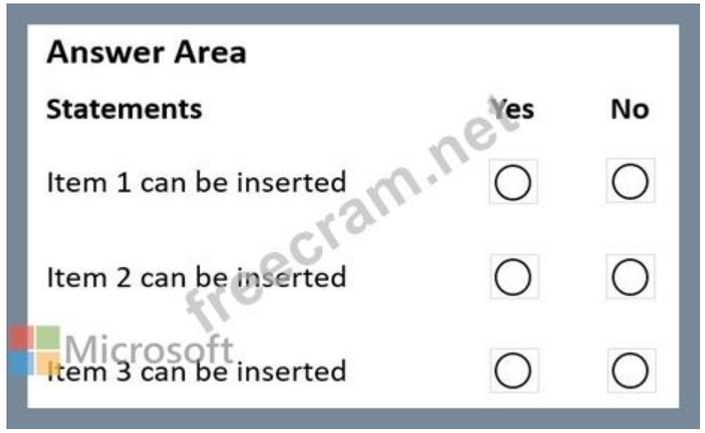
Contacts contains the items shown in the following table.

companyID	familyName	givenNames	emailAddress
1	Rogers	Terry John	terryj@contoso.com
1	Rogers	null	rogers@contoso.com

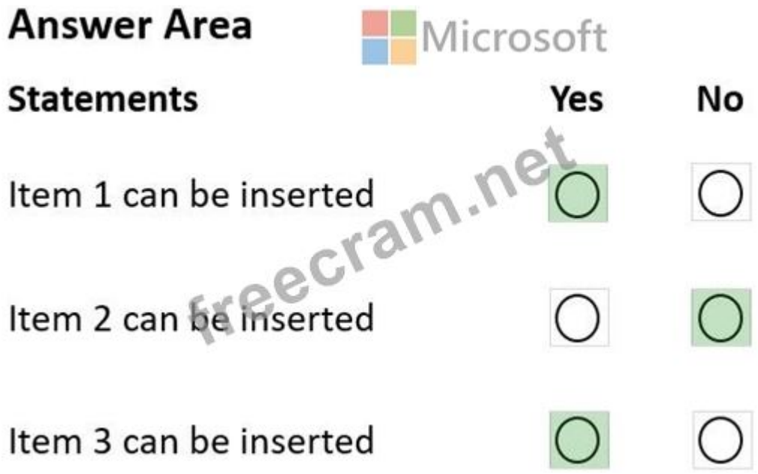
To Contacts, you plan to Insert the items shown in the following table.

Item	companyID	familyName	givenNames	emailAddress
1	1	Rogers	terry john	terryj@contoso.com
2	1	Rogers	null	rogers2@contoso.com
3	2	Rogers	Terry John	terryj@contoso.com

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:



NEW QUESTION: 16

Hotspot Question

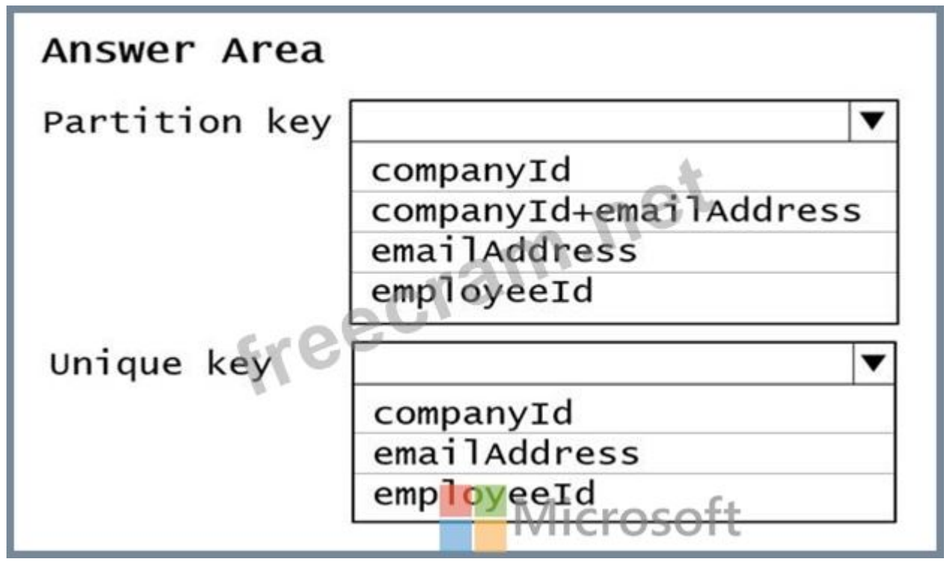
You have a database in an Azure Cosmos DB Core (SQL) API account.

You plan to create a container that will store employee data for 5,000 small businesses. Each business will have up to 25 employees. Each employee item will have an emailAddress value.

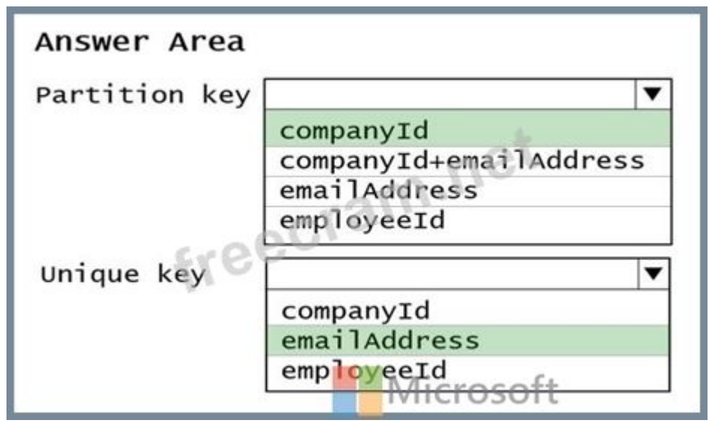
You need to ensure that the emailAddress value for each employee within the same company is unique.

To what should you set the partition key and the unique key? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.



Answer:



Explanation:

Box 1: CompanyID

After you create a container with a unique key policy, the creation of a new or an update of an existing item resulting in a duplicate within a logical partition is prevented, as specified by the unique key constraint. The partition key combined with the unique key guarantees the uniqueness of an item within the scope of the container.

For example, consider an Azure Cosmos container with Email address as the unique key constraint and CompanyID as the partition key. When you configure the user's email address with a unique key, each item has a unique email address within a given CompanyID. Two items can't be created with duplicate email addresses and with the same partition key value.

Box 2: emailAddress

UniqueKey will need to be set to emailAddress, since the job of a unique key is to ensure uniqueness of a value within a logical partition.

Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/unique-keys

Valid DP-420 Dumps shared by EduDump.com for Helping Passing DP-420 Exam! EduDump.com now offer the **newest DP-420 exam dumps**, the EduDump.com DP-420 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com DP-420 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/DP-420/premium/ (**281** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 17

Hotspot Question

You have an Azure subscription that contains an Azure Cosmos DB for NoSQL database named DB1. The shared throughput provisioned for DB1 is 10,000 DTU/s. DB1 contains the containers shown in the following table.

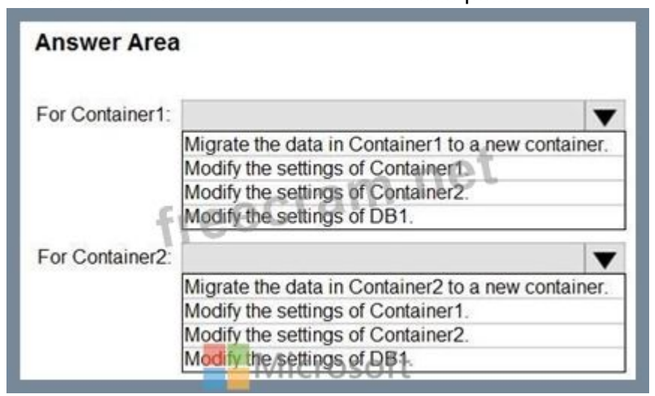
Name	Provisioned throughput
Container1	Share throughput across containers
Container2	4,000 DTU/s

You need to modify the throughput for the containers. The solution must meet the following requirements:

- The maximum throughput for Container1 must be 4,000 DTU/s.
- The throughput for Container2 must be shared across the containers.
- Administrative effort must be minimized.

What should you do? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.



Answer:

Answer Area

For Container1:
- Migrate the data in Container1 to a new container.
- **Modify the settings of Container1.**
- Modify the settings of Container2.
- Modify the settings of DB1.

For Container2:
- **Migrate the data in Container2 to a new container.**
- Modify the settings of Container1.
- Modify the settings of Container2.
- Modify the settings of DB1.

NEW QUESTION: 18

Note: This section contains one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem. You must determine whether the solution meets the stated goals. More than one solution in the set might solve the problem. It is also possible that none of the solutions in the set solve the problem.

After you answer a question in this section, you will NOT be able to return. As a result, these questions do not appear on the Review Screen.

You have an Azure Cosmos DB for NoSQL database. The database hosts a container named families that stores data in the following format.

```
{
  "id": "WakefieldFamily",
  "parents": [
    { "familyName": "Wakefield", "givenName": "Robin" },
    { "familyName": "Miller", "givenName": "Ben" }
  ],
  "children": [
    {
      "familyName": "Merriam",
      "givenName": "Jesse",
      "gender": "female", "grade": 1,
      "pets": [
        { "type": "dog", "givenName": "Goofy" },
        { "type": "cat", "givenName": "Shadow" }
    ]
  },
    {
      "familyName": "Miller",
      "givenName": "Lisa",
      "gender": "female",
      "grade": 8
    }
  ],
  "address": { "state": "NY", "county": "Manhattan", "city": "NY" },
  "creationDate": 1431620462,
  "isRegistered": false
}
```

freecram.net
Microsoft

You need to return the following data:

- The ID of the family
- The family names of the children
- The name of any pets if the pet is a cat

Solution: You run the following query.

```
...
SELECT
  f.id,
  c.familyName AS FamilyName,
  p.givenName AS petName
FROM
  families f
JOIN
  f.children c
JOIN
  (SELECT value p FROM c.pets p WHERE p.type = 'cat') AS p
...
```

Microsoft
freecram.net

Does this meet the goal?

A. Yes

B. No

Answer: (SHOW ANSWER)

Correct:

* You run the following query.

```
...
SELECT
  f.id,
  c.familyName AS FamilyName,
  p.givenName AS petName
FROM
  families f
JOIN
  f.children c
JOIN
  (SELECT value p FROM c.pets p WHERE p.type = 'cat') AS p
...
```

Microsoft
freecram.net

The following Azure Cosmos DB NoSQL query fulfills the requirements by unnesting child array items and filtering the pet array:

To retrieve the family ID, the children's family names, and the rabbit's name, you can use the query below.

SELECT f.id, c.familyName AS FamilyName, p.givenName AS petName

FROM families f

JOIN f.children c

JOIN (SELECT VALUE p FROM p IN c.pets WHERE p.type = 'cat') AS p

1. Flatten the Children Array

To access individual properties within the children sub-array, use a JOIN statement. In Azure Cosmos DB, a JOIN performs an inner product between the document root (f) and the array (f.children). This filters out any family documents that contain no children.

2. Isolate Cat Pets

To inspect the properties of the pets array embedded inside each child, apply a secondary JOIN on c.pets. The WHERE clause then filters these projected items to only include cases where the property $p.type$ matches the string "cat".

3. Project Specific Attributes

The SELECT clause restricts the final JSON output schema to contain exactly three custom- aliased properties:

familyId: Extracted from the root document

IDchildFamilyName: Extracted from the current child item

iterationpetName: Extracted from the matching pet item iteration

Incorrect:

* You run the following query.

```
...
SELECT
  f.id,
  p.familyName AS FamilyName,
  pet.givenName AS petName
FROM
  families f
JOIN
  f.parents p
JOIN
  f.children c
JOIN
  (SELECT value pet FROM c.pets pet WHERE pet.type = 'cat') AS pet
...
```

Use two JOINS and one subquery, not three JOINS and a subquery.

* You run the following query.

```
...
SELECT
  f.id,
  c.familyName AS FamilyName,
  p.givenName AS petName
FROM
  families f
CROSS APPLY
  f.children c
CROSS APPLY
  c.pets p
WHERE
  p.type = 'cat'
...
```

CROSS APPLY should not be used.

Reference:

https://learn.microsoft.com/en-us/cosmos-db/query/subquery

NEW QUESTION: 19

You have an Azure Cosmos DB for NoSQL account.

You configure the diagnostic settings to send all log information to a Log Analytics workspace.

You need to identify when the provisioned request units per second (RU/s) for resources within the account were modified.

You write the following query.

```
AzureDiagnostics
| where Category == "ControlPlaneRequests"
```

What should you include in the query?

A. | where OperationName startswith "SqlContainersCreate"

B. | where OperationName startswith "SqlContainersDelete"

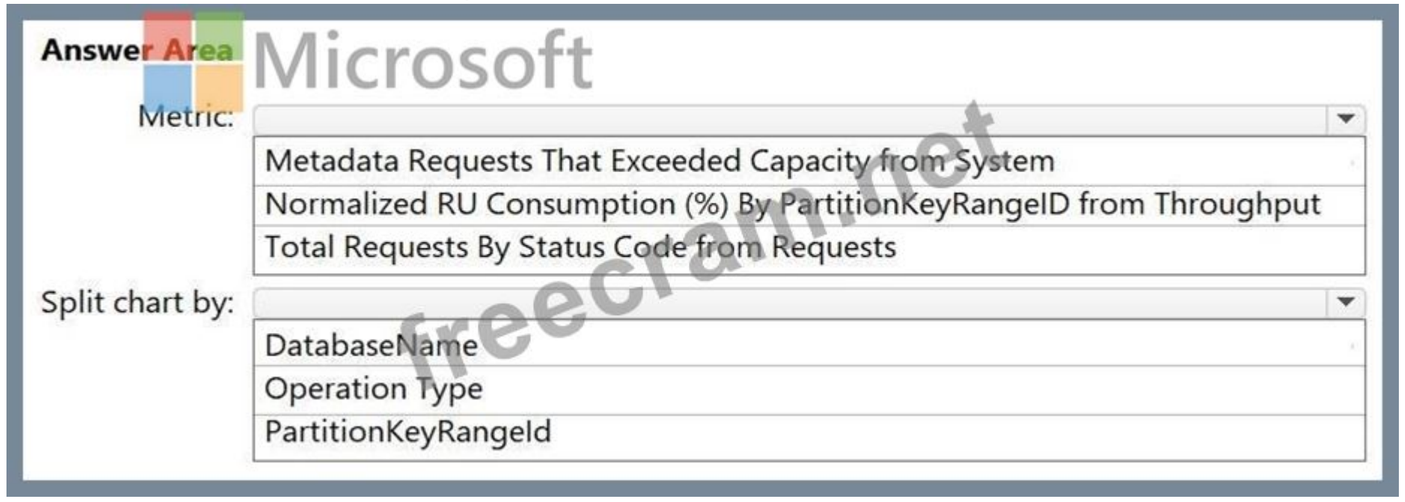
C. | where OperationName startswith "SqlDatabasesThroughputUpdate"
D. | where OperationName startswith "AccountUpdateStart"
Answer: (SHOW ANSWER)
In Azure Cosmos DB, the provisioned request units per second (RU/s) can be modified at the container (SQL Container) or database level. The operation name for modifying RU/s is captured under "SqlDatabasesThroughputUpdate" in the diagnostic logs.

NEW QUESTION: 20
You have an Azure Cosmos DB for NoSQL container named Container1 that stores 500 GB of data.
You plan to run cross-partition queries that target Container1. The queries will include the partition key values specified in the db filters.
You need to minimize the cost of the queries.
What should you do?
Freecram.net
A. Decrease the value of the MaxBufferedItemCount parameter.
B. Add the DISTINCT keyword to the queries.
C. Decrease the value of the MaxConcurrency parameter.
D. Add the in keyword to the queries.
Answer: (SHOW ANSWER)
To keep the price (RU cost) of your queries low in an Azure Cosmos DB for NoSQL container with hundreds of GB of data, the correct action is: Add the IN keyword to the queries.
Why it works: Using the IN keyword allows you to provide a specific list of partition key values in the query's WHERE clause.
Cost reduction: By specifying partition keys, the query engine can route the request only to the specific physical partitions that hold those values.
Efficiency: This avoids a full "fan-out" query (visiting every physical partition in a container with hundreds of GBs), which is the most expensive way to query Cosmos DB.
Incorrect:
[Not A]
Decrease the value of the MaxBufferedItemCount parameter:
This parameter controls how many items the client-side SDK buffers during a cross-partition query to manage memory and latency. It does not reduce the Request Unit (RU) cost charged by the database.
[Not B]
Add the DISTINCT keyword to the queries:
The DISTINCT keyword often increases the RU cost because the query engine must perform extra work to identify and remove duplicate values across multiple partitions.
[Not C]
Decrease the value of the MaxConcurrency parameter:
MaxConcurrency limits the number of simultaneous network requests the SDK makes to different partitions. While it can help manage client-side resources and avoid rate limiting, it does not reduce the total RU cost of the query execution on the server side.
Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/performance-tips-query-sdk

NEW QUESTION: 21
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 (Vnet)
D. You can use RBAC (role-based access control) to authenticate data plane requests routed through the dedicated gateway
Answer: (SHOW ANSWER)

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

NEW QUESTION: 22
Hotspot Question
You have an Azure Cosmos DB for NoSQL account named account1. account1 hosts a container named Container1 that has provisioned throughput.
You have an app named App1 that uses Container1.
You use Azure Monitor to review the performance and capacity of account1.
Users report intermittent request failures during peak traffic, and the logs for App1 show an HTTP
429 Too many requests error.
You need to identify whether a hot physical partition is causing uneven throughput consumption.
What should you use in Azure Cosmos DB Insights? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:



NEW QUESTION: 23

Hotspot Question

You are migrating a relational database to Azure Cosmos DB for NoSQL. The existing database contains the following relational tables.

Name	Type	Related to	Number of rows
Customer	Entity	*None*	14,500
BusinessCategory	Reference	Customer	45
SalesTerritory	Reference	Customer	14
BuyingGroup	Reference	Customer	27
Product	Entity	*None*	8,700
ProductType	Reference	Product	38
Brand	Reference	Product	193
Color	Reference	Product	14
SalesOrderItem	Entity	Customer, Product	45,000,000

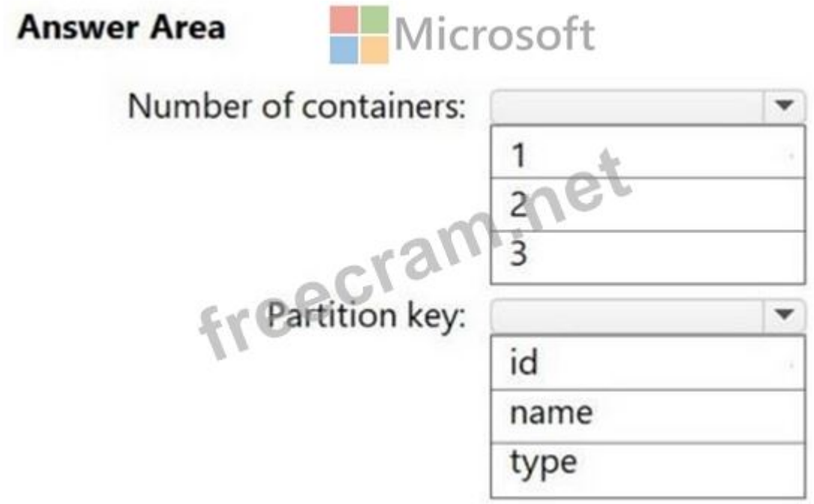
Microsoft
freeCram.net

The most commonly executed query for each reference table is to return all the items in the reference table.

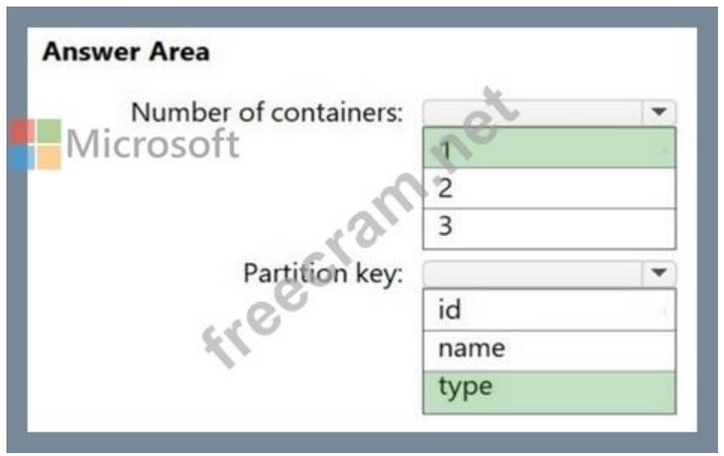
You need to migrate all the reference tables to Azure Cosmos DB. The solution must minimize the number of containers.

How should you configure storage for the reference tables? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.



Answer:



NEW QUESTION: 24

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

You need to develop a model that supports read operations for orders. The solution must minimize the number of requests.

What should you do?

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

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

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

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

Answer: (SHOW ANSWER)

By denormalizing data, your application may need to issue fewer queries and updates to complete common operations.

Typically denormalized data models provide better read performance.

Note: In general, use embedded data models when:

There are contained relationships between entities.
There are one-to-few relationships between entities.
There's embedded data that changes infrequently.
There's embedded data that will not grow without bound.
There's embedded data that is queried frequently together.
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/sql/modeling-data

NEW QUESTION: 25
You have a container named container1 in an Azure Cosmos DB for NoSQL 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: (SHOW ANSWER)
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.
Incorrect Answers:
A: checkpointInterval: When set, it defines, in milliseconds, the interval between lease checkpoints.
Default is always after each Function call.
C: maxItemsPerInvocation: When set, this property sets the maximum number of items received per Function call. If operations in the monitored collection are performed through stored procedures, transaction scope is preserved when reading items from the change feed. As a result, the number of items received could be higher than the specified value so that the items changed by the same transaction are returned as part of one atomic batch.
Reference: https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-cosmosdb- v2-trigger

NEW QUESTION: 26
You have a database in an Azure Cosmos DB for NoSQL account.
You need to create an Azure function that will access the database to retrieve records based on a variable named accountnumber. The solution must protect against SQL injection attacks.
How should you define the command statement in the function?
A. cmd = "SELECT * FROM Persons p
WHERE p.accountnumber = 'accountnumber'"
B. cmd = "SELECT * FROM Persons p
WHERE p.accountnumber = LIKE @accountnumber"
C. cmd = "SELECT * FROM Persons p
WHERE p.accountnumber = @accountnumber"
D. cmd = "SELECT * FROM Persons p
WHERE p.accountnumber = '" + accountnumber + "'"

Answer: (SHOW ANSWER)
Azure Cosmos DB supports queries with parameters expressed by the familiar @ notation.
Parameterized SQL provides robust handling and escaping of user input, and prevents accidental exposure of data through SQL injection.
For example, you can write a query that takes lastName and address.state as parameters, and execute it for various values of lastName and address.state based on user input.
SELECT *
FROM Families f
WHERE f.lastName = @lastName AND f.address.state = @addressState
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/sql/sql-query-parameterized- queries

NEW QUESTION: 27
Hotspot Question
You have an Azure Cosmos DB for NoSQL database named DB1 that hosts a container named Container1. The partition key for Container1 is /city.
You are developing a cloud-native app named App1 that will store customer information in DB1.
App1 will perform transactional batch operations to update the items in Container1.
You need to configure App1 to meet the following requirements:
- Ensure that batch operations are rolled back if an item's ETag value
has changed since the value was last read.
- Ensure that all operations in the batch apply to a single partition
key value.
How should you complete the C# code? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
using Microsoft.Azure.Cosmos;
using System;
using System.Threading.Tasks;
namespace CosmosBatch
{
  public class Program
  {
    public static async Task Main(string[] args)
    {
      CosmosClient client = new CosmosClient("<account endpoint>", "<account key>");
      Container container = client.GetContainer("db1", "container1");
      // Ensure transaction applies to a single partition
      TransactionalBatch batch = await batch.ExecuteAsync(new TransactionalBatchRequestOptions [ ▼ ]
      batch.UpsertItem(new { id = "1", city = "London" });
      // Ensure rollback if ETag value changes
      container.CreateTransactionalBatch(new PartitionKey [ ▼ ]
    }
  }
}
```

- { IfMatchEtag = "etag1" });
- { IfNoneMatchEtag = "etag1" });
- { IfNoneMatchEtag = "etag1", EnableContentResponseOnWrite = false });

- ("city"));
- ("/city"));
- ("London"));

freecram.net Microsoft

Answer:

Answer Area

```
using Microsoft.Azure.Cosmos;
using System;
using System.Threading.Tasks;
namespace CosmosBatch
{
  public class Program
  {
    public static async Task Main(string[] args)
    {
      CosmosClient client = new CosmosClient("<account endpoint>", "<account key>");
      Container container = client.GetContainer("db1", "container1");
      // Ensure transaction applies to a single partition
      TransactionalBatch batch = await batch.ExecuteAsync(new TransactionalBatchRequestOptions
      batch.UpsertItem(new { id = "1", city = "London" });
      // Ensure rollback if ETag value changes
      container.CreateTransactionalBatch(new PartitionKey
    }
  }
}
```

- { IfMatchEtag = "etag1" });
- { IfNoneMatchEtag = "etag1" });
- { IfNoneMatchEtag = "etag1", EnableContentResponseOnWrite = false });

- ("city"));
- ("/city"));
- ("London"));

freecram.net
Microsoft

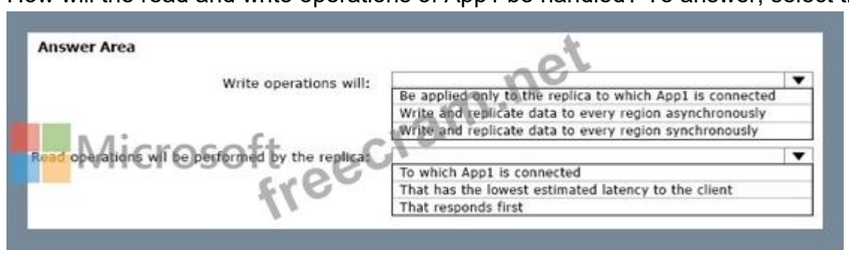
NEW QUESTION: 28

Hotspot Question

You have a multi-region Azure Cosmos DB account named account1 that has a default consistency level of strong.

You have an app named App1 that is configured to request a consistency level of session.

How will the read and write operations of App1 be handled? To answer, select the appropriate options in the answer area.



Answer:

Answer Area

Write operations will:
- Be applied only to the replica to which App1 is connected
- Write and replicate data to every region asynchronously
- Write and replicate data to every region synchronously

Read operations wil be performed by the replica:
- To which App1 is connected
- That has the lowest estimated latency to the client
- That responds first

Explanation:
An account configured for strong consistency by default will still write and replicate data synchronously to every region in the account. When the SDK client instance or request overrides this with Session or weaker consistency, reads will be performed using a single replica.
https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels

NEW QUESTION: 29
You are designing an Azure Cosmos DB for NoSQL 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: (SHOW ANSWER)
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.
Incorrect Answers:
A: You will also not like to partition the data on "DateTime", because this will create a hot partition. Imagine you have partitioned the data on time, then for a given minute, all the calls will hit one partition. If you need to

retrieve the data for a customer, then it will be a fan-out query because data may be distributed on all the partitions.
B: Senser1Value has only two values.
C: All the devices could have the same manufacturer.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/sql/synthetic-partition-keys

NEW QUESTION: 30
Hotspot Question
You have an Azure Cosmos DB for NoSQL database named db1.
You run the following .NET code.

```
...
Database database = client.GetDatabase("db1");
ContainerProperties properties = new ()
{
    Id = "Container1",
    PartitionKeyPath = "/customerId",
    DefaultTimeToLive = 3600,
    UniqueKeyPolicy = new UniqueKeyPolicy
    {
        UniqueKeys = new Collection<UniqueKey>(
            new List<UniqueKey>
            {
              new UniqueKey
                {
                  Paths = new Collection<string>(
                    new List<string>
                      {
                        "/firstName",
                        "/lastName",
                        "/emailAddress"
                      }
                    )
                },
            }
        )
    }
};
```

freecram.net
Microsoft

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:



NEW QUESTION: 31

You plan to deploy an Azure Cosmos DB for NoSQL account that will contain orders and the order lines of each order.

You need to recommend an account design that will maximize performance and minimize the cost of reading a single order and its order lines.

What should you recommend?

A. Embed the order lines of each order in the same container.

B. Store the orders in a separate container from the order lines.

C. Store individual orders and order lines as separate items in the same container. Normalize the items to minimize their size.

D. Store individual orders and individual order lines as separate items in the same container. Include a type attribute in each item.

Answer: (SHOW ANSWER)

To optimize for high performance and low cost when reading a single order and its order lines together, you should embed the order lines as an array within the order document.

Single Request: You fetch the order and all lines in one ReadItemAsync point read.

Lowest Latency: Point reads are the fastest operation in Cosmos DB.Lowest Cost: Reading a single document costs exactly 1 Request Unit (RU) for data under 1 KB.

No Joins: Cosmos DB does not support cross-document joins efficiently without burning RUs on queries.

Atomic Updates: You can update the order and its lines together safely in one operation.

How to Structure the Datajson{

"id": "order_1001",

"partitionKey": "customer_9942",

"orderDate": "2026-06-03T17:27:00Z",

"status": "Shipped",

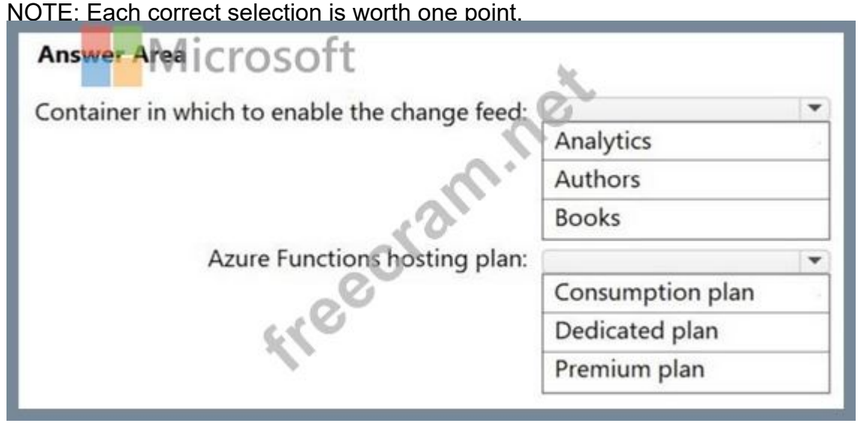
"orderLines": [
{ "lineId": 1, "productId": "prod_abc", "quantity": 2, "price": 15.99 },
{ "lineId": 2, "productId": "prod_xyz", "quantity": 1, "price": 45.00 }
]
}
Reference:
https://devblogs.microsoft.com/cosmosdb/read-data-spring-azure-cosmos-db-v3/

Valid DP-420 Dumps shared by EduDump.com for Helping Passing DP-420 Exam! EduDump.com now offer the **newest DP-420 exam dumps**, the EduDump.com DP-420 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com DP-420 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/DP-420/premium/ (**281** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

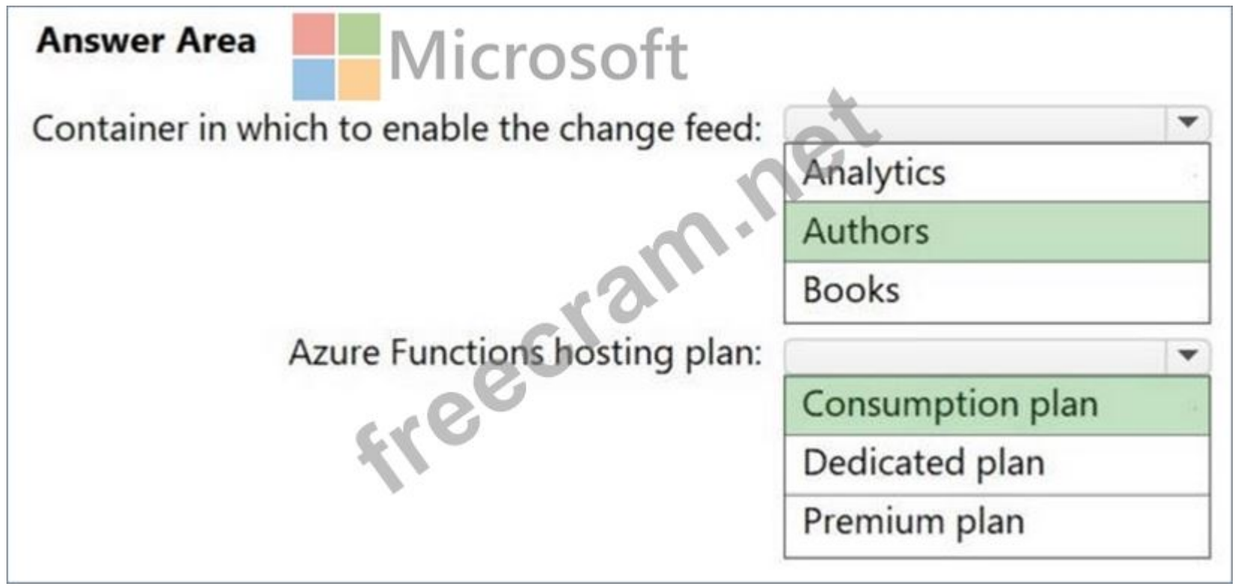
NEW QUESTION: 32
Hotspot Question
You have an Azure Cosmos DB for NoSQL account that hosts the containers shown in the following table.

Name	Description
Authors	The container stores personal details, biographies, and links to profile images of authors.
Books	The container stores properties, marketing details, and links to cover images of books. Each item includes an `authorId` and an `authorName` field.
Analytics	The container stores aggregated book sales data. Each item includes an `authorId` field.

You need to implement a solution for processing changes to author names. The solution must use a change feed and an Azure Function and must meet the following requirements:
- Ensure that when a name change occurs, the authorName field in the
Books container is updated within 30 minutes.
- Minimize costs.
What should you configure? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:



freecram.net

NEW QUESTION: 33

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

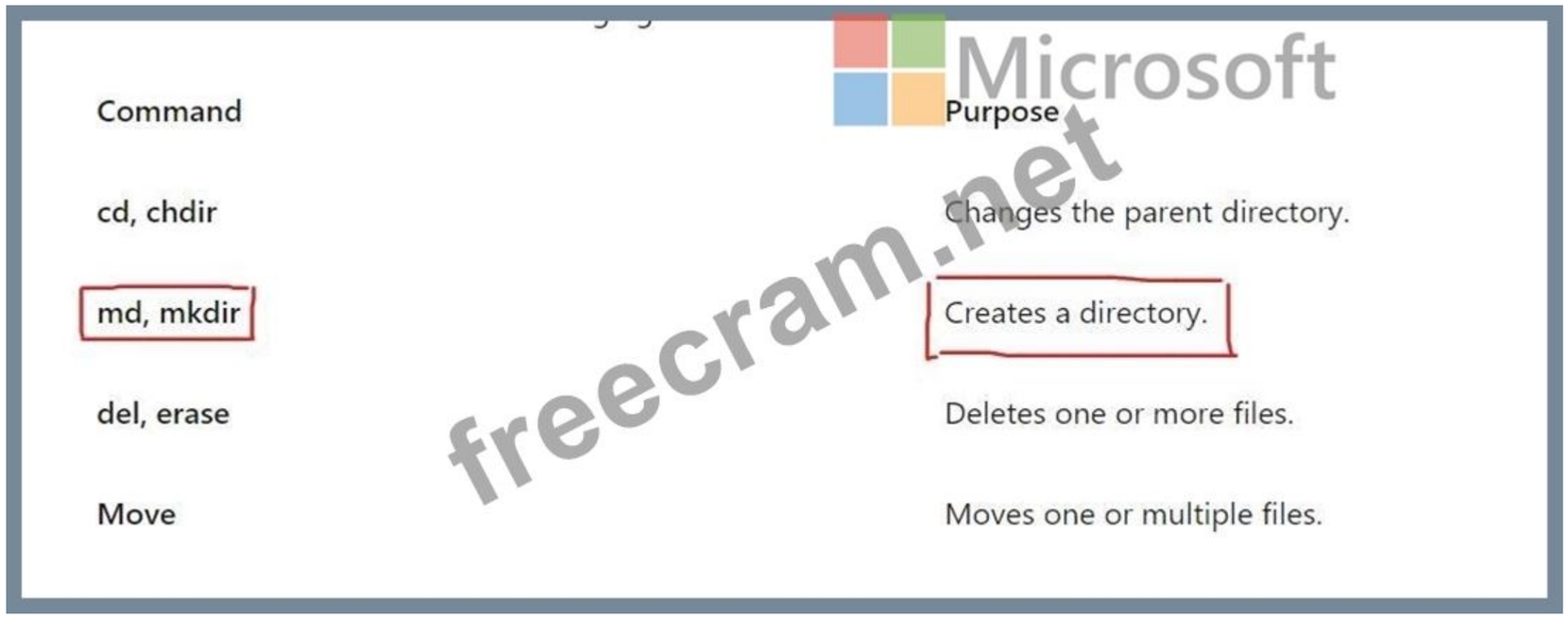
C. Cassandra

D. Gremlin

E. Table

Answer: (SHOW ANSWER)

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



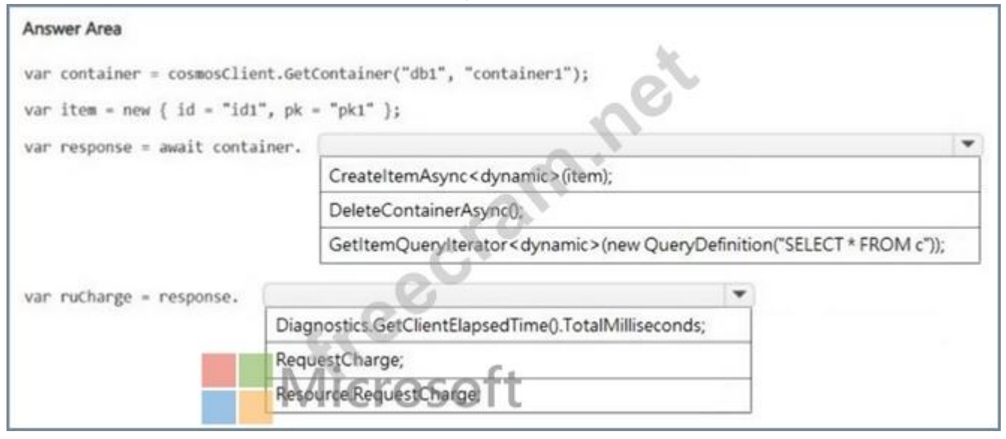
Microsoft
freecram.net

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

NEW QUESTION: 34
You have an Azure Cosmos DB for NoSQL account.
You configure the diagnostic settings to send all log information to a Log Analytics workspace.
You need to identify when the provisioned request units per second (RU/s) for resources within the account were modified.
You write the following query.
AzureDiagnostics
| where Category == "ControlPlaneRequests"
What should you include in the query?
A. | where OperationName startswith " AccountUpdateComplete"
B. | where OperationName startswith "SqlContainersCreate"
C. | where OperationName startswith "MongoCollectionsThroughputUpdate"
D. | where OperationName startswith "SqlContainersThroughputUpdate"
Answer: (SHOW ANSWER)
Correct:
* . | where OperationName startswith "SqlContainersThroughputUpdate"

In Azure Cosmos DB, the provisioned request units per second (RU/s) can be modified at the container (SQL Container) or database level. The operation name for modifying RU/s is captured under "SqlDatabasesThroughputUpdate" in the diagnostic logs.
To identify when the provisioned request units (RU/s) were modified, you should filter for operations related to throughput updates. The operation name SqlContainersThroughputUpdate specifically indicates that a change has been made to the throughput settings for a container in a Cosmos DB account. This allows you to track modifications effectively.
Incorrect:
* | where OperationName startswith "AccountUpdateStart"
* | where OperationName startswith " AccountUpdateComplete"
* | where OperationName startswith "MongoCollectionsThroughputUpdate"
* | where OperationName startswith "RegionAddComplete"
* | where OperationName startswith "SqlContainersCreate"
* | where OperationName startswith "SqlContainersDelete"
Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/request-units

NEW QUESTION: 35
Hotspot Question
You have an Azure Cosmos DB for NoSQL database named DB1 that hosts a container named Container1. The partition key for Container1 is /pk1.
You have a .NET app that connects to DB1.
You need to retrieve the request unit (RU) cost of a point operation and store the value as a variable.
How should you complete the C# code? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:



Microsoft
freecram.net

NEW QUESTION: 36
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: (SHOW ANSWER)
If you create two different types of documents including the property 'deviceid' for both entities, it will ease referencing both entities inside the container.

NEW QUESTION: 37
You have an Azure Cosmos DB for NoSQL account named account1.
In account1, you plan to create a container named container1 that will contain data continuously collected from self-driving cars. The items in container1 will be accessed mostly by using queries that target data collected from a specific car during a given day. Each car is identifiable by using a property named Vehicle-Identification-Number.
You need to identify a partition strategy for container1. The strategy must meet the following requirements:
- Maximize read and write performance.
- Minimize the costs of data reads and writes.
What should you identify?
A. Use a partition key based on the Vehicle-Identification-Number property.
B. Construct a synthetic key by concatenating the current date and a random suffix.
C. Use a partition key based on the current date.
D. Construct a synthetic key by concatenating the current date and a hash of the Vehicle- Identification-Number property value.

Answer: (SHOW ANSWER)
For an Azure Cosmos DB for NoSQL container storing continuous self-driving car data, the optimal partition strategy is to use the requested synthetic partition key constructed by concatenating the current date and a hash of the Vehicle-Identification-Number (VIN).
Why this strategy is optimal:
Maximizes Read Performance:
Since your primary access pattern targets a specific car on a given day, this synthetic key ensures that all relevant data for that query resides in a single logical partition.
This allows the database to perform efficient in-partition queries rather than expensive and slow cross-partition scans.
Maximizes Write Performance:
Including a hash of the VIN ensures high cardinality, which prevents "hot partitions".
Even if many cars are reporting data simultaneously, the hash-based distribution spreads these writes across multiple physical partitions.
Minimizes Costs:
By avoiding cross-partition queries, you significantly reduce Request Unit (RU) consumption, as each physical partition scan adds overhead (typically 2-3 RUs per partition).
The 20-GB logical partition limit is less likely to be exceeded because data for a single car is scoped to a single day rather than its entire history.
Reference:
https://www.dclessons.com/azure-cosmos-db

NEW QUESTION: 38
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account.
You need to write a SQL query that will return multiple items. The solution must meet the following requirements:
- For each item returned, include the current Coordinated Universal Time (UTC) date and time, formatted as an ISO 8601 string.
- Minimize development effort.
What should you use?
A. a user-defined function
B. GetCurrentTimestamp()
C. GetCurrentDateTime()
D. DateTimePart()
Answer: (SHOW ANSWER)
GetCurrentTimestamp() is a built-in system function in Azure Cosmos DB's SQL API, which returns the current Coordinated Universal Time (UTC) in ISO 8601 format as a string. It can be used directly in a SQL query without any additional effort. This approach minimizes development effort because it doesn't require custom logic or user-defined functions.

NEW QUESTION: 39
You have a container in an Azure Cosmos DB for NoSQL 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.

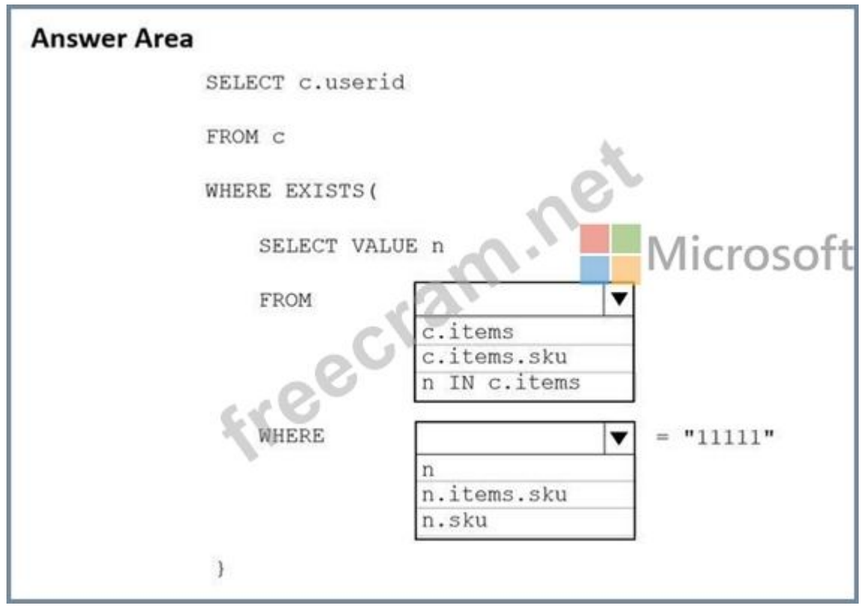
```
{
    "id" : "9ccf1906-2a30-4dc0-9644-2185f5dcbbd7",
    "deviceId" : "bba6fe24-6d97-4935-8d58-36baa4b8a0e1",
    "telemetryId" : "9d7816e6-f401-42ba-ad05-0e03de35c0b8",
    "date" : "2019-05-03",
    "time" : "13:05",
    "temp" : "21"
}
```

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: (SHOW ANSWER)
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

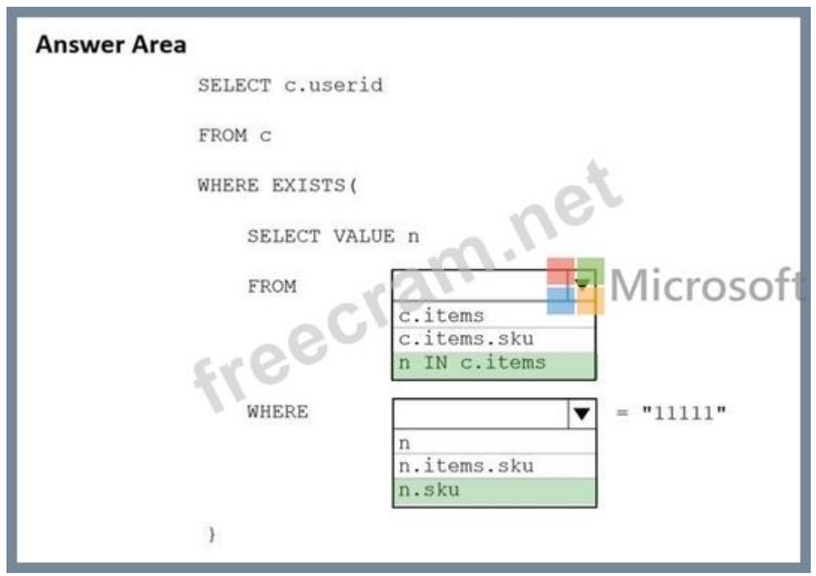
NEW QUESTION: 40
Hotspot Question
You plan to create an Azure Cosmos DB for NoSQL container that will contain items in the following format.

```
"userid": "user1@contoso.com",
"id": "e379aea5-63f5-4623-92e9b-4cd9b33b91d5",
"orderdate": "2022-11-01702:00:00",
"items": [
    {
        "sku": "11111",
        "qty": 1
    },
    {
        "sku": "11112",
        "qty": 5
    }
]
```

You need to define a query that will return the userid values of all the users who purchased an item that has a sku value of 11111.
How should you compete the query? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:



NEW QUESTION: 41

You have a container named container1 in an Azure Cosmos DB for NoSQL account named account1 that is set to the session default consistency level. The average size of an item in contained is 20 KB.

You have an application named App1 that uses the Azure Cosmos DB SDK and performs a point read on the same set of items in container1 every minute.

You need to minimize the consumption of the request units (RUs) associated to the reads by App1.

What should you do?

A. In App1, modify the connection policy settings

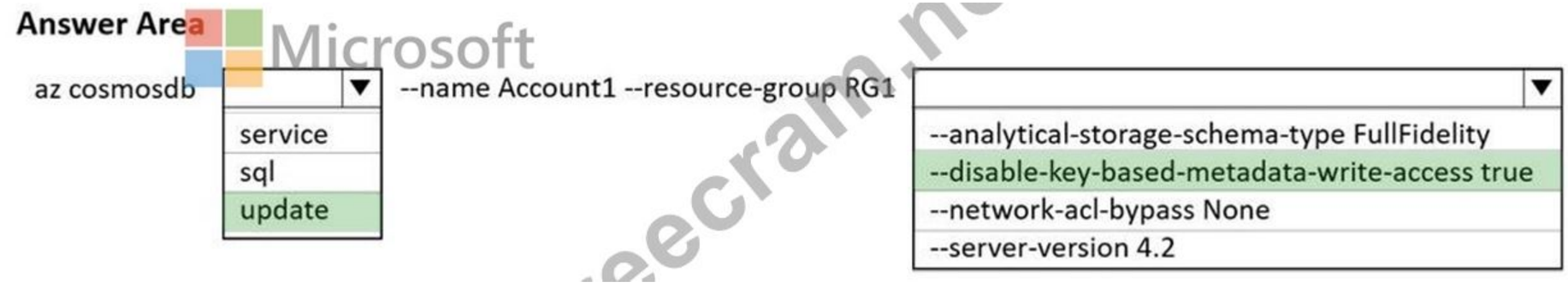
B. In App1, change the consistency level of read requests to consistent prefix

C. In account1, provision a dedicated gateway and integrated cache
D. In account1, change the default consistency level to bounded staleness
Answer: (SHOW ANSWER)
The main goal of the integrated cache is to reduce costs for read-heavy workloads. Low latency, while helpful, is not the main benefit of the integrated cache because Azure Cosmos DB is already fast without caching. Point reads and queries that hit the integrated cache won't use any RUs. In other words, any cache hits will have an RU charge of 0. Cache hits will have a much lower per-operation cost than reads from the backend database.
Workloads that fit the following characteristics should evaluate if the integrated cache will help lower costs:
- Read-heavy workloads
- Many repeated point reads on large items
- Many repeated high RU queries
- Hot partition key for reads
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/integrated-cache

NEW QUESTION: 42
Hotspot Question
You have an Azure subscription that contains a resource group named RG1. RG1 contains an Azure Cosmos DB for NoSQL account named account1.
You plan to audit changes to the control plane of account1.
You need to ensure that audit events include the details of the security principal that performs each control plane change.
How should you complete the az command? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:



NEW QUESTION: 43
You have an Azure Cosmos DB for NoSQL account that contains a database named DB1.
Users run the following query against DB1.
SELECT *
FROM c
WHERE c.bikeParts = "Frame, Chain, Peddles"
ORDER BY c.bikeParts ASC
You add an index to the recordset.
You need to change the ORDER BY syntax of the query to optimize the use of the index. The solution must NOT change the result set.
Which syntax should you use?
A. ORDER BY c._ts ASC, c.bikeParts DESC
B. ORDER BY c._ts DESC, c.bikeParts ASC
C. ORDER BY c.bikeParts ASC, c._ts ASC
D. ORDER BY c.bikeParts DESC, c._ts ASC
Answer: (SHOW ANSWER)
Need to first sort on c.bikeParts DESC to keep the same order.
Note: ts. A read-only and system-generated timestamp. The value is the Document's last-updated timestamp, in seconds from the UNIX epoch.
Incorrect:
[Not A, Not C]
Need ASC c.bikeParts sorting on not DESC
[Not B]
Need to first sort on c.bikeParts to keep the same order.
If the indexing mode is set to consistent, the system properties id and _ts are automatically indexed.
Reference:
https://docs.safe.com/fme/html/FME-Form-Documentation/FME-ReadersWriters/documentdb/feature-representation.htm

NEW QUESTION: 44
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL database named db1. The subscription is linked to a Microsoft Entra tenant that contains a user named User1.
You need to delegate permissions to User1 to meet the following requirements:
- Ensure that User1 can manage db1.
- Prevent User1 from accessing connection strings.
- Prevent User1 from accessing the data stored in db1.
Which role should you assign to User1?
A. Cosmos DB Account Reader Role
B. Cosmos DB Built-in Data Contributor
C. Cosmos DB Built-in Data Reader
D. Cosmos DB Operator
Answer: (SHOW ANSWER)
To ensure the user can only manage the database without accessing connection strings or stored data, you should assign the Cosmos DB Operator role.
This role is specifically designed for the control plane, allowing administrative tasks while strictly isolating data-plane access.
Role Details and Permissions
Management Capabilities: The user can manage Azure Cosmos DB accounts, databases, and containers.

Access Restrictions: By design, this role cannot view or regenerate account keys (connection strings).
Data Isolation: This role does not grant permissions to read, write, or query the actual data stored within the containers.
Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/how-to-connect-role-based-access-control

NEW QUESTION: 45
Hotspot Question
You have a development environment that uses multiple Azure subscriptions.
You plan to deploy a single Azure Cosmos DB account that will contain a single database in each subscription.
You need to automate the deployment by using an Azure Resource Manager (ARM) template.
The solution must meet the following requirements:
- Minimize Azure Cosmos DB-related charges.
- Maximize the available request units per second (RU/s) without
increasing costs.
How should you complete the template? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
...
  "resources": [
{
  "type": "Microsoft.DocumentDB/databaseAccounts",
  "apiVersion": "2022-05-15",
  "name": "[toLower(parameters('accountName'))]",
  "location": "[parameters('location')]",
  "properties": {
```

"connectorOffer":'small'
"enableFreeTier": true,
"kind":"GlobalDocumentDB"

```
    "databaseAccountOfferType": "Standard",
    "consistencyPolicy": {
    "defaultConsistencyLevel": "Session"
  },
  "locations": [
    {
```

```
        "locationName": "[parameters('location')]"
...
  {
    "type": "Microsoft.DocumentDB/databaseAccounts/sqlDatabases",
    "apiVersion": "2022-05-15",
    "name": "[format('{0}/{1}', toLower(parameters('accountName')), parameters('databaseName'))]",
    "properties": {
      "resource": {
        "id": "[parameters('databaseName')]"
      },
      "options": {
```

"throughput": 400
"throughput": 1000
"throughput": 2000

```
      }
...
      "dependsOn": [
        "[resourceId('Microsoft.DocumentDB/databaseAccounts', toLower(parameters('accountName')))]"
      ...
}
```

Answer:

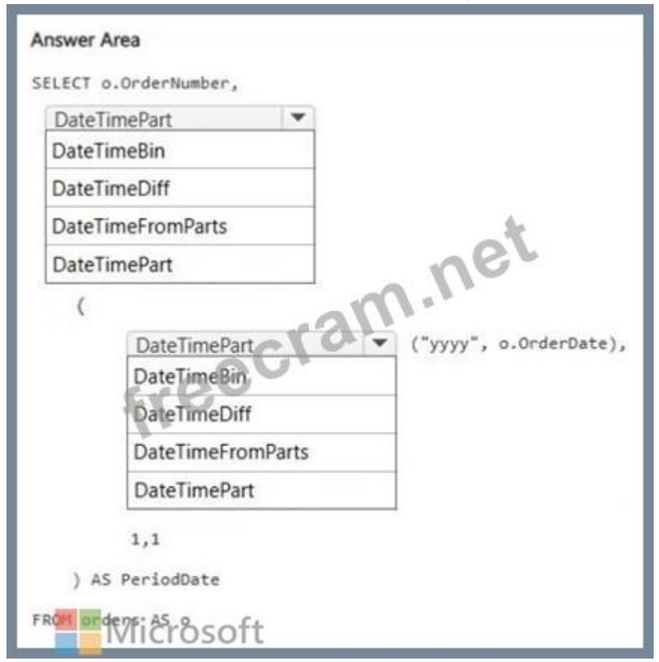
Answer Area

```
...
  "resources": [
    {
      "type": "Microsoft.DocumentDB/databaseAccounts",
      "apiVersion": "2022-05-15",
      "name": "[toLower(parameters('accountName'))]",
      "location": "[parameters('location')]",
```

```
properties : {
    [ "connectorOffer":'small' | "enableFreeTier": true, | "kind":"GlobalDocumentDB" ]
    "databaseAccountOfferType": "Standard",
    "consistencyPolicy": {
    "defaultConsistencyLevel": "Session"
  },
  "locations": [
    {
      "locationName": "[parameters('location')]"
...
 {
   "type": "Microsoft.DocumentDB/databaseAccounts/sqlDatabases",
   "apiVersion": "2022-05-15",
   "name": "[format('{0}/{1}', toLower(parameters('accountName')), parameters('databaseName'))]",
   "properties": {
     "resource": {
       "id": "[parameters('databaseName')]"
     },
     "options": {
       [ "throughput": 400 | "throughput": 1000 | "throughput": 2000 ]
     }
...
     "dependsOn": [
       "[resourceId('Microsoft.DocumentDB/databaseAccounts', toLower(parameters('accountName')))]"
     ...
}
```

freecram.net

NEW QUESTION: 46
Hotspot Question
You have an Azure Cosmos DB for NoSQL database that hosts a container named Container1.
Container1 stores order details, including the following properties:
- OrderNumber (a number)
- OrderDate (a date)
You need to write a query that returns the following for each order:
- OrderNumber
- PeriodDate (as the first day of the calendar year based on orderDate) How should you complete the query? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point



freecram.net



freecram.net
Microsoft

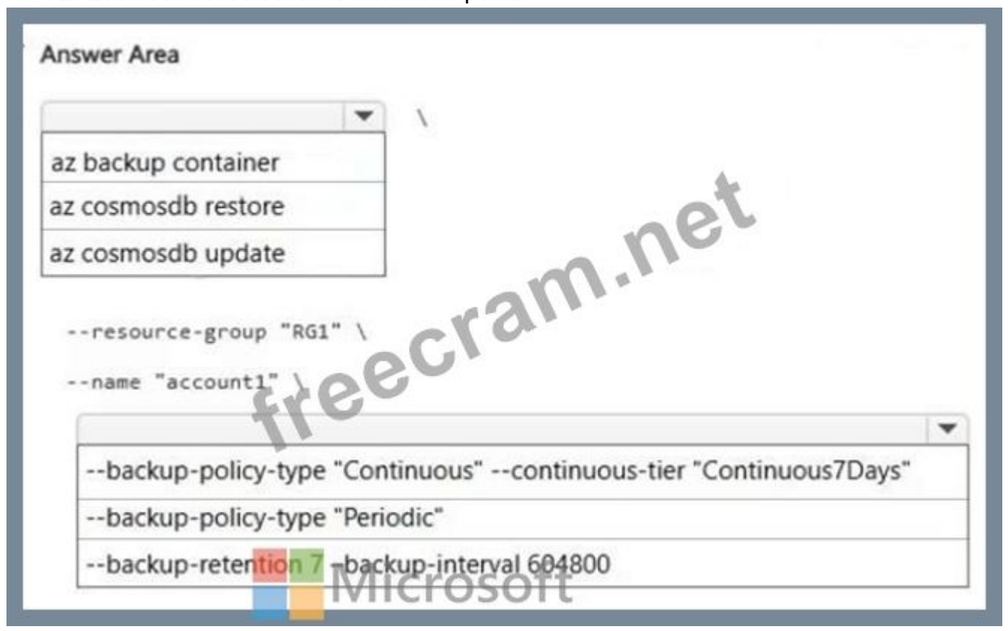
Valid DP-420 Dumps shared by EduDump.com for Helping Passing DP-420 Exam! EduDump.com now offer the **newest DP-420 exam dumps**, the EduDump.com DP-420 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com DP-420 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/DP-420/premium/ (**281** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 47
You have an Azure Cosmos DB for NoSQL account. The account hosts a container that has the change feed enabled.
You are building an app by using the Azure Cosmos DB SDK. The app will read items from the change feed by using a pull model.
You need to ensure that multiple threads can read the change feed in parallel.
What should you include?
A. the ChangeFeedStartFrom value
B. the pagesize value
C. the FeedRange value
D. a stream-based iterator
Answer: (SHOW ANSWER)
To enable parallel processing of the change feed with multiple threads, you should use the FeedRange value in the Azure Cosmos DB SDK. The FeedRange allows partitioning of the change feed, enabling multiple threads to process different ranges of the feed in parallel. This is essential for scaling the processing of changes across multiple threads or even multiple instances of an application.

NEW QUESTION: 48

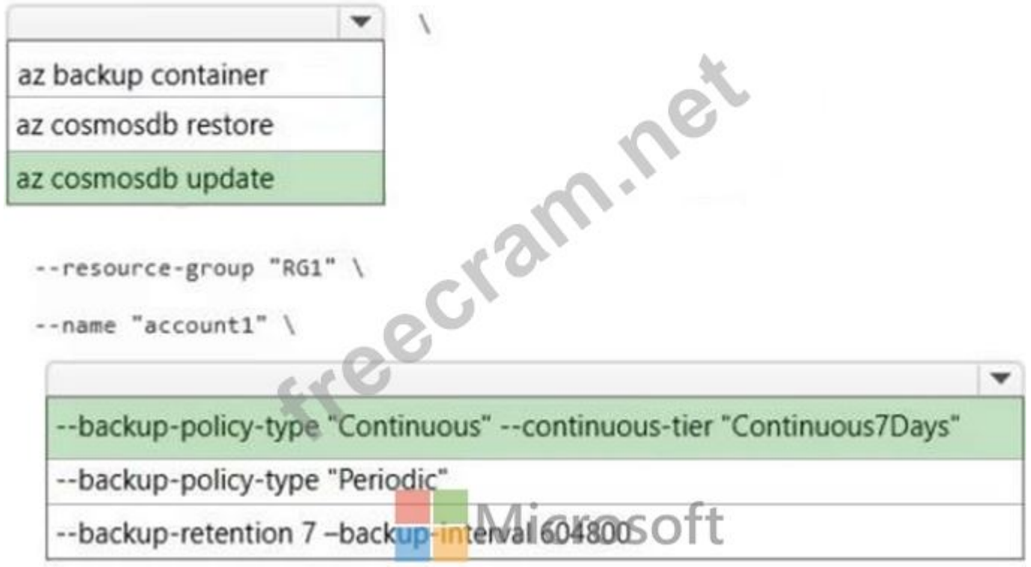
While chairing a team session, you are telling the team members about the Azure Cosmos DB Emulator.
Which of the following statement is not true about Azure Cosmos DB Emulator?
A. Azure Cosmos DB Emulator offers an emulated environment that would run on the local developer workstation.
B. The emulator 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: (SHOW ANSWER)
https://docs.microsoft.com/en-us/azure/cosmos-db/local-emulator

NEW QUESTION: 49
Hotspot Question
You have an Azure Cosmos DB for NoSQL account named account1.
You need to ensure that account1 will support self-service point-in-time restore for the next seven days.
How should you complete the Azure Command-Line Interface (CLI) script? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



freecram.net
Microsoft
Answer:

Answer Area



NEW QUESTION: 50

Case Study 1 - Litware, inc

Overview

Litware, Inc. is a United States-based grocery retailer. Litware has a main office and a primary datacenter in Seattle. The company has 50 retail stores across the United States and an emerging online presence. Each store connects directly to the internet. Existing environment.

Cloud and Data Service Environments. Litware has an Azure subscription that contains the resources shown in the following table.

Name	Type	Description
account1	Azure Cosmos DB for NoSQL account	The account has a single read-write region and contains a database named productdb. The productdb database contains two containers named con-product and con-productVendor. The account uses the session default consistency level.
iothub1	Azure IoT hub	The IoT hub collects per-second temperature and humidity telemetry from 5,000 IoT devices in the retail stores. A single telemetry reading generates 1 KB of data.
streamanalytics1	Azure Stream Analytics job	The job processes telemetry data collected from iothub1.

Each container in productdb is configured for manual throughput. The con-product container stores the company's product catalog data. Each document in con-product includes a con- productvendor value. Most queries targeting the data in con-product are in the following format.

SELECT * FROM con-product p WHERE p.con-productVendor - 'name'

Most queries targeting the data in the con-productVendor container are in the following format SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Existing environment.

Current Problems.

Litware identifies the following issues:

- Updates to product categories in the con-productVendor container do not propagate automatically to documents in the con-product container.

- Application updates in con-product frequently cause HTTP status code 429 "Too many requests".
- You discover that the 429 status code relates to excessive request unit (RU) consumption during the updates.

Requirements.

Planned Changes

Litware plans to implement a new Azure Cosmos DB for NoSQL account named account2 that will contain a database named iotdb. The iotdb database will contain two containers named con- iot1 and con-iot2.

Litware plans to make the following changes:

- Store the telemetry data in account2.
- Configure account1 to support multiple read-write regions.
- Implement referential integrity for the con-product container.
- Use Azure Functions to send notifications about product updates to different recipients.
- Develop an app named App1 that will run from all locations and query the data in account1.
- Develop an app named App2 that will run from the retail stores and query the data in account2.
- App2 must be limited to a single DNS endpoint when accessing account2.

Requirements.

Business Requirements

Litware identifies the following business requirements:

- 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.
- Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.
- Minimize the number of firewall changes in the retail stores.

Requirements.

Product Catalog Requirements

Litware identifies the following requirements for the product catalog:

- Implement a custom conflict resolution policy for the product catalog data.
- Minimize the frequency of errors during updates of the con-product container.
- Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.
- Trigger the execution of two Azure functions following every update to any document in the con- product container.

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 (LEAVE A REPLY)

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: 51

Hotspot Question

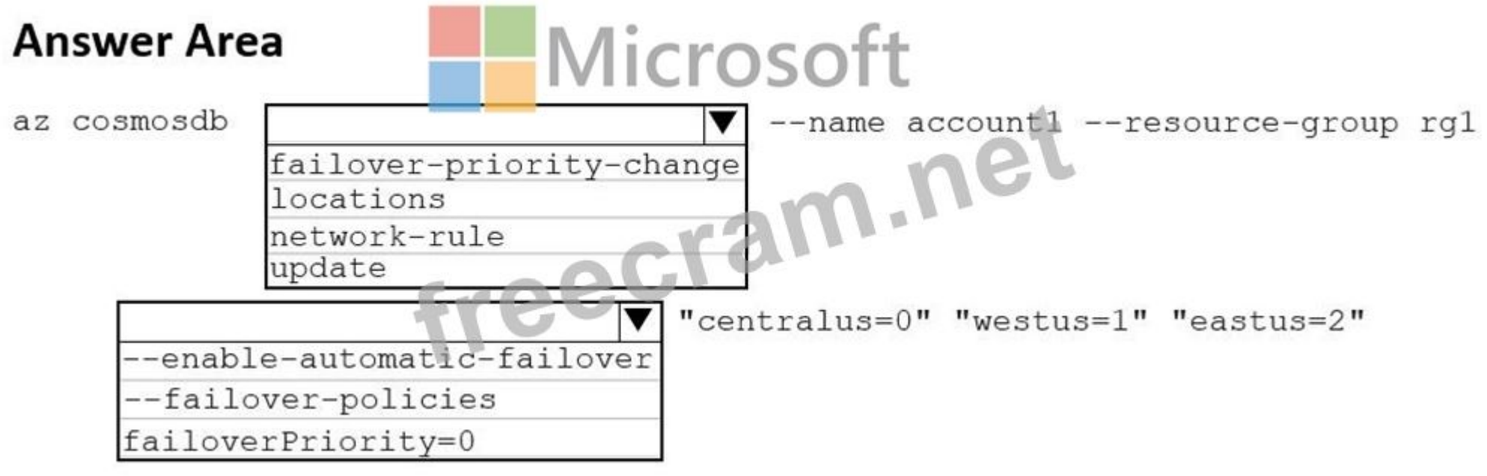
You have an Azure Cosmos DB for NoSQL account named account1 in a resource group named RG1. The Azure regions for account1 are shown in the following table.

Name	Type
East US (eastus)	Write
Central US (centralus)	Read
West US (westus)	Read

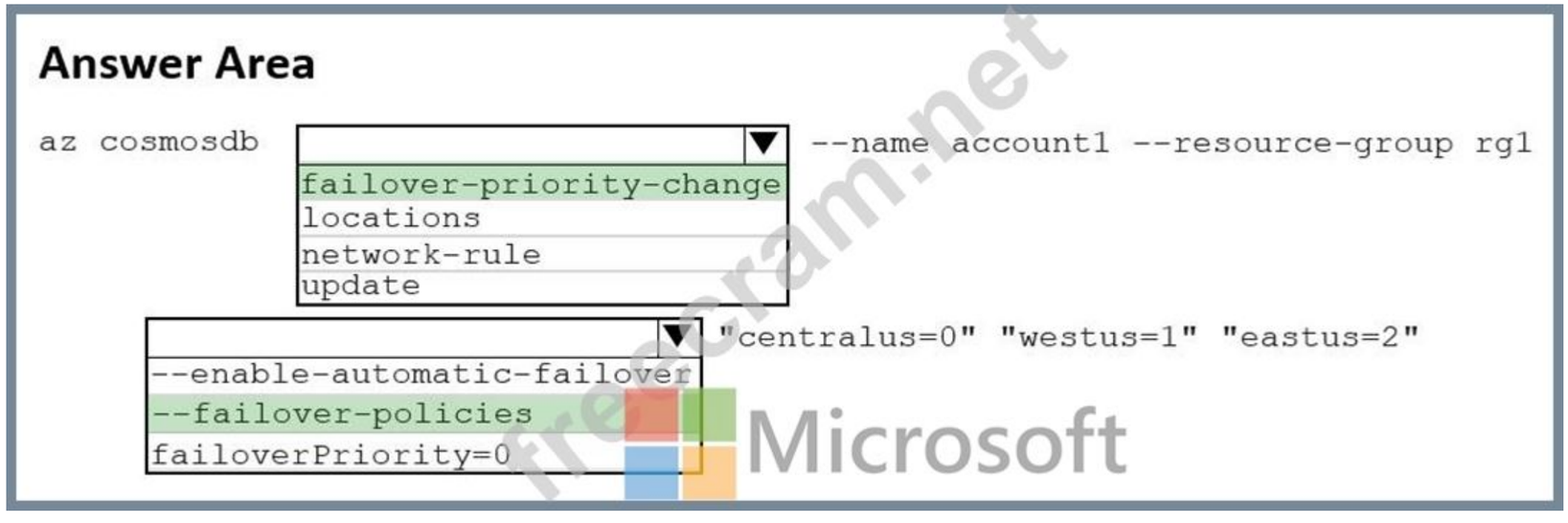
You need to fail over account1 from the East US region to the Central US region by using Azure Command-Line Interface (CLI).

How should you complete the command? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.



Answer:



NEW QUESTION: 52

Hotspot Question

You have an Azure Cosmos DB for NoSQL account that hosts a container named Container1.

Container1 stores items in the following format.

```
{
  "id": "e379aea5-63f5-4623-9a9b-4cd9b33b91d5",
  "name": "R-410 Road Bicycle",
  "inventory": {
    "quantity": 15
  },
  "color": "silver"
}
```

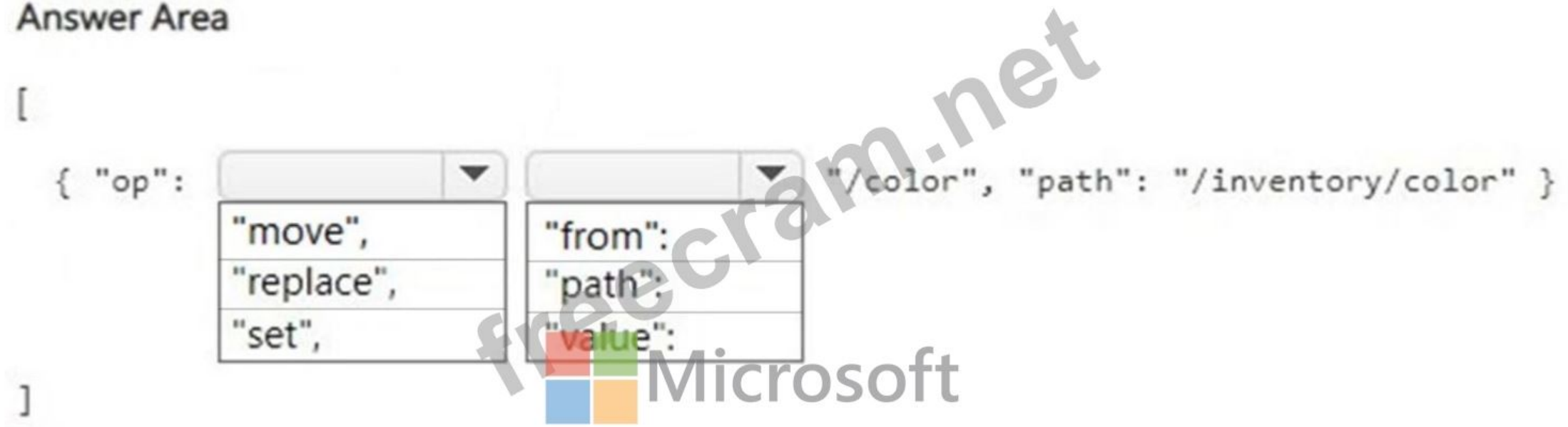
You need to modify the contents of Container1 to conform to the following format.

```
{
  "id": "e379aea5-63f5-4623-9a9b-4cd9b33b91d5",
  "name": "R-410 Road Bicycle",
  "inventory": {
    "quantity": 15,
    "color": "silver"
  }
}
```

How should you complete the JSON Patch document? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Microsoft
freecram.net



Answer:



NEW QUESTION: 53

You have a database named db1 in an Azure Cosmos DB for NoSQL account named account1.

You need to write JSON data to db1 by using Azure Stream Analytics. The solution must minimize costs.

Which should you do before you can use db1 as an output of Stream Analytics?

A. In account1, add a private endpoint

B. In db1, create containers that have a custom indexing policy and analytical store disabled
C. In db1, create containers that have an automatic indexing policy and analytical store enabled
D. In account1, enable a dedicated gateway
Answer: (SHOW ANSWER)
Stream Analytics doesn't create containers in your database. Instead, it requires you to create them beforehand. You can then control the billing costs of Azure Cosmos DB containers. You can also tune the performance, consistency, and capacity of your containers directly by using the Azure Cosmos DB APIs.
https://learn.microsoft.com/en-us/azure/stream-analytics/stream-analytics-documentdb-output#basics-of-azure-cosmos-db-as-an-output-target

NEW QUESTION: 54
You are designing an Azure Cosmos DB for NoSQL solution to store data from IoT devices.
Writes from the devices will be occur every second. Data will be retained indefinitely.
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"
}
```

freecram.net
Microsoft

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: (SHOW ANSWER)
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.
Incorrect Answers:
A: You will also not like to partition the data on "DateTime", because this will create a hot partition. Imagine you have partitioned the data on time, then for a given minute, all the calls will hit one partition. If you need to retrieve the data for a customer, then it will be a fan-out query because data may be distributed on all the partitions.
B: Senser1Value has only two values.
C: All the devices could have the same manufacturer.

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

NEW QUESTION: 55
You have an Azure Cosmos DB account named account1 that contains a database named DB1 and a container name Container1. account1 uses the Core (SQL) API.
You have an app named App1 that stores a large volume of data in Container1. The data is partitioned by using the int1 attribute. The items stored in Container1 have the following format.

```
{
  "id": "unique id",
  "int1": "partition key value",
  "string1": "string1",
  "string2": "string2",
  "string3": "string3",
  "string4": "string4",
  "string5": "string5"
}
```

App1 runs the following query.

```
SELECT c.string1,c.string2,c.string3
FROM container1 c
WHERE c.int1 = some_value
```

You need to recommend a solution to minimize the required request units (RUs) when performing read and write operations. What should you recommend?
A. Use the default indexing policy.
B. For DB1, set Throughput to Autoscale.
C. Configure the indexing policy to include only int1 and exclude string1, string2, and string3.
D. Configure the indexing policy to include only int1 and exclude all other attributes.
Answer: (SHOW ANSWER)
To reduce Request Units (RUs) for both read and write operations, you should include only the attributes used in the query filter and exclude all unrequested properties from your indexing policy.
You should include int1, string1, and string2 in your indexing policy paths, and exclude all other paths (including string3, string4, and string5).
Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/understand-request-unit-consumption

NEW QUESTION: 56
Hotspot Question
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account. The account contains a container named Container1 that stores product data and stock level information.
You need to implement a stored procedure that will accept the following parameters:
- productid
- productName
- initialStockLevel
The solution must create the following:
- An item that contains the product data
- An item that contains the stock level data of each product

How should you complete the stored procedure? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
function addNewProduct(productID, productName, initialStockLevel) {
  var dbContext = getContext();
  var container = dbContext.
```

- getCollection()
- getETag()
- getID()
- getResourceID()
- getSelfLink()

```
  var product = {
    id: productID,
    type: "product",
    name: productName
  };
  var productStockLevel = {
    id: productID,
    type: "productStockLevel",
    currentStockLevel: initialStockLevel
  };
  var containerLink = container.
```

- getETag()
- getID()
- getResourceID()
- getSelfLink()

```
  var productAccepted = container.createDocument(containerLink, product,
    function (productError, addedProduct) {
      if (productError) throw new Error('Error' + productError.message);
```

freecram.net

```
    var productStockLevelAccepted = container.createDocument(containerLink, productStockLevel,
      function (productStockLevelError, addedProductStockLevel) {
        if (productStockLevelError) throw new Error('Error' + productStockLevelError.message);
        context.getResponse().setBody({
          product: addedProduct,
          stocklevel: addedProductStockLevel
...
}
```

Microsoft

Answer:

Answer Area

```
function addNewProduct(productID, productName, initialStockLevel) {
  var dbContext = getContext();
  var container = dbContext.
```

- getCollection()
- getETag()
- getID()
- getResourceID()
- getSelfLink()

```
var product = {
  id: productID,
  type: "product",
  name: productName
};
var productStockLevel = {
  id: productID,
  type: "productStockLevel",
  currentStockLevel: initialStockLevel
};
var containerLink = container.
```

- getETag()
- getID()

Microsoft
freecram.net

```
                                    getResourceID()
                                    getSelfLink()
  var productAccepted = container.createDocument(containerLink, product,
    function (productError, addedProduct) {
      if (productError) throw new Error('Error' + productError.message);
      var productStockLevelAccepted = container.createDocument(containerLink, productStockLevel,
        function (productStockLevelError, addedProductStockLevel) {
          if (productStockLevelError) throw new Error('Error' + productStockLevelError.message);
          context.getResponse().setBody({
            product: addedProduct,
            stocklevel: addedProductStockLevel
...
}
```

NEW QUESTION: 57

You have an Azure Cosmos DB for NoSQL database named DB1 that hosts the containers shown in the following table.

Name	Description
Container1	Stores user information
Container2	Stores user permissions
Container3	Stores product information
Container4	Stores product price information

The user permissions are bound to a user and will NOT be queried independently of the user.

The product price is bound to a product and will NOT be queried independently of the product.

You are designing a new data model for DB1.

You need to optimize the data model to improve read performance and maintainability. The solution must meet the following requirements:

- Future enhancements of the user permissions must be supported.
- Future enhancements of the product price do NOT need to be supported.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Create a new container to store both the product information and the product price in separate documents and link the documents by using a shared product ID.

B. Denormalize the product price into product documents in Container3.

C. Denormalize the user permissions into user documents in Container1.

D. Denormalize user documents into the user permissions in Container2.

E. Denormalize product documents into the product price in Container4.

F. Store both the user information and the user permissions in a single container in separate documents that have references.

Answer: B,C (LEAVE A REPLY)

To improve read performance and maintainability, you must transition from a normalized relational design to an optimized NoSQL design. NoSQL databases prioritize reading data in a single request rather than joining multiple containers.

Container Denormalization Strategy

[B]

Product Data (Container3 and Container4)

Action: Denormalize Container4 into Container3 by flattening the fields.

Product prices are never queried independently. Combining them allows a single read operation to fetch the complete product details.

Maintainability: Since future price enhancements are not required, you can directly flatten the price fields (e.g., price, currency) into the root of the product document. This keeps the schema simple and highly efficient.

[C]

User Data (Container1 and Container2)

Action: Denormalize Container2 into Container1 using reference (embedding as an array).

Users and permissions are always queried together. Embedding eliminates a second network call, drastically improving read performance.

Maintainability: Because future enhancements to permissions must be supported, embedding them as a strongly-typed JSON array or sub-object within the user document allows you to easily add new fields (e.g., expiration dates, scopes) later without breaking the schema.

Reference:

https://microsoftlearning.github.io/dp-420-cosmos-db-dev/instructions/17-denormalize.html

NEW QUESTION: 58

Hotspot Question

You have an Azure Cosmos DB for NoSQL account named Account1 that uses provisioned throughput capacity mode.

Account1 contains the databases shown in the following table.

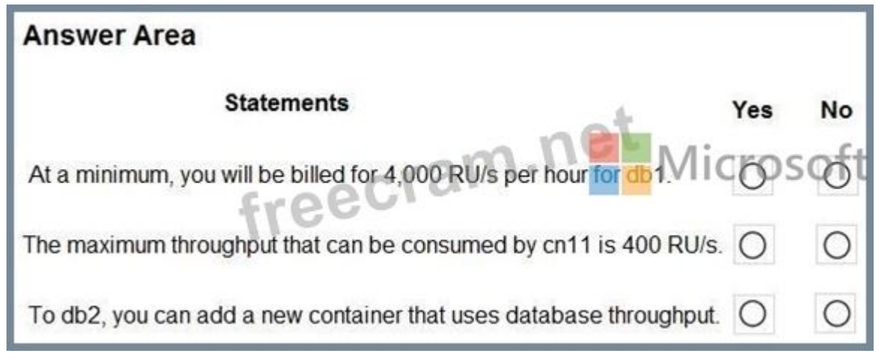
Name	Throughput	Max request units per second (RU/s)	Geo-redundancy	Multi-region writes	Number of regions
db1	Autoscale	5,000	Disabled	Disabled	1
db2	Autoscale	8,000	Enabled	Enabled	3

The databases contain the containers shown in the following table.

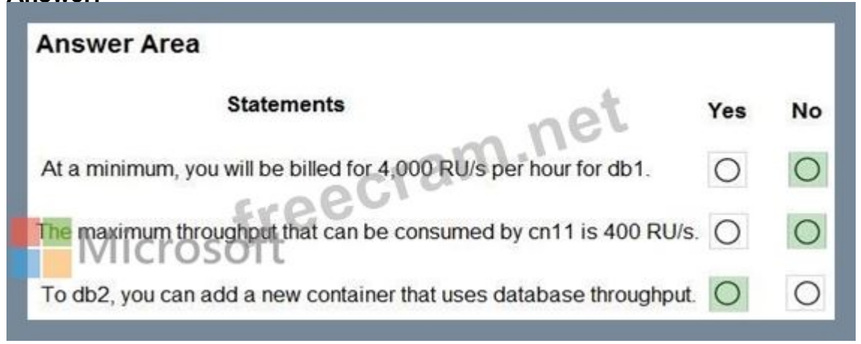
Name	Database	Throughput
cn01	db1	Container – autoscale maximum RU/s of 10,000
cn02	db1	Database
cn03	db1	Database
cn04	db1	Database
cn05	db1	Database
cn11	db2	Database
cn12	db2	Database
cn13	db2	Database
cn14	db2	Database
cn15	db2	Database
cn16	db2	Database
cn17	db2	Database
cn18	db2	Database

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:



NEW QUESTION: 59

You have an Azure Cosmos DB for NoSQL account that has multiple write regions.

You need to receive an alert when requests that target the database exceed the available request units per second (RU/s).

Which Azure Monitor signal should you use?

A. Data Usage

B. Metadata Requests

C. Region Removed

D. Region Added

Answer: B (LEAVE A REPLY)

Metric Metadata Requests: Count of metadata requests. Azure Cosmos DB maintains system metadata container for each account, that allows you to enumerate collections, databases, etc., and their configurations, free of charge.

Used to monitor throttles due to metadata requests.

Incorrect:

Not A: Metric Data Usage: Total data usage reported at 5-minutes granularity per region. Used to monitor total data usage at container and region, minimum granularity should be 5 minutes.

Reference:

https://docs.microsoft.com/en-us/azure/cosmos-db/monitor-cosmos-db-reference#metrics

NEW QUESTION: 60

You are implementing an Azure Data Factory data flow that will use an Azure Cosmos DB for NoSQL sink to write a dataset. The data flow will use 2,000 Apache Spark partitions.
You need to ensure that the ingestion from each Spark partition is balanced to optimize throughput.
Which sink setting should you configure?

Freecram.net

A. Throughput
B. Write throughput budget
C. Batch size
D. Collection action

Answer: (SHOW ANSWER)

Batch size: An integer that represents how many objects are being written to Cosmos DB collection in each batch. Usually, starting with the default batch size is sufficient. To further tune this value, note: Cosmos DB limits single request's size to 2MB. The formula is "Request Size = Single Document Size * Batch Size". If you hit error saying "Request size is too large", reduce the batch size value.
The larger the batch size, the better throughput the service can achieve, while make sure you allocate enough RUs to empower your workload.
Incorrect Answers:
A: Throughput: Set an optional value for the number of RUs you'd like to apply to your CosmosDB collection for each execution of this data flow. Minimum is 400.
B: Write throughput budget: An integer that represents the RUs you want to allocate for this Data Flow write operation, out of the total throughput allocated to the collection.
D: Collection action: Determines whether to recreate the destination collection prior to writing.
None: No action will be done to the collection.
Recreate: The collection will get dropped and recreated
Reference: https://docs.microsoft.com/en-us/azure/data-factory/connector-azure-cosmos-db

NEW QUESTION: 61

Hotspot Question
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account. The account hosts a container named Container1 that stores customer data. Each customer data item includes the customer's name and zero or more related business categories in an array.
You need identify customers that have one of their business categories set to "retail". The solution must return the customerName and businessCategories fields.
How should you complete the query? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
SELECT
  c.customerName,
  c.businessCategories
FROM
 customers c
WHERE
```

[▼] (SELECT [▼] bc

ARRAY_CONTAINS
BETWEEN
DISTINCT
EXISTS
LIKE
VALUE

ARRAY_CONTAINS
BETWEEN
DISTINCT
EXISTS
LIKE
VALUE

```
FROM bc IN c.businessCategories
WHERE bc = "retail")
```

Microsoft
freecram.net

Answer:

Answer Area

```
SELECT
  c.customerName,
  c.businessCategories
FROM
 customers c
WHERE
```

[▼] (SELECT [▼] bc

ARRAY_CONTAINS	ARRAY_CONTAINS
BETWEEN	BETWEEN
DISTINCT	DISTINCT
EXISTS	EXISTS
LIKE	LIKE
VALUE	VALUE

```
FROM bc IN c.businessCategories
WHERE bc = "retail")
```

freecram.net

Valid DP-420 Dumps shared by EduDump.com for Helping Passing DP-420 Exam! EduDump.com now offer the **newest DP-420 exam dumps**, the EduDump.com DP-420 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com DP-420 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/DP-420/premium/ (**281** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 62
Drag and Drop Question
You have an Azure Cosmos DB container named owners. Each item in owners represents an owner and their pets.
The following is a sample of the JSON

```
{
    "id": "1",
    "ownerName": "Fred",
    "pets": [
        {
            "petName": "Fluffy",
            "age": 1
        },
        {
            "petName": "Bouncy",
            "age": 3
        },
        {
            "petName": "Hairy",
            "age": 10
        }
    ]
}
```

Microsoft
freecram.net

You are writing a query that will retrieve the average age of each pet owned by each owner, if the pet is older than two years.

How should you complete the query? To answer, drag the appropriate values to the correct targets. Each value may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Values

AS

COUNT

IN

JOIN

SUM

VALUE

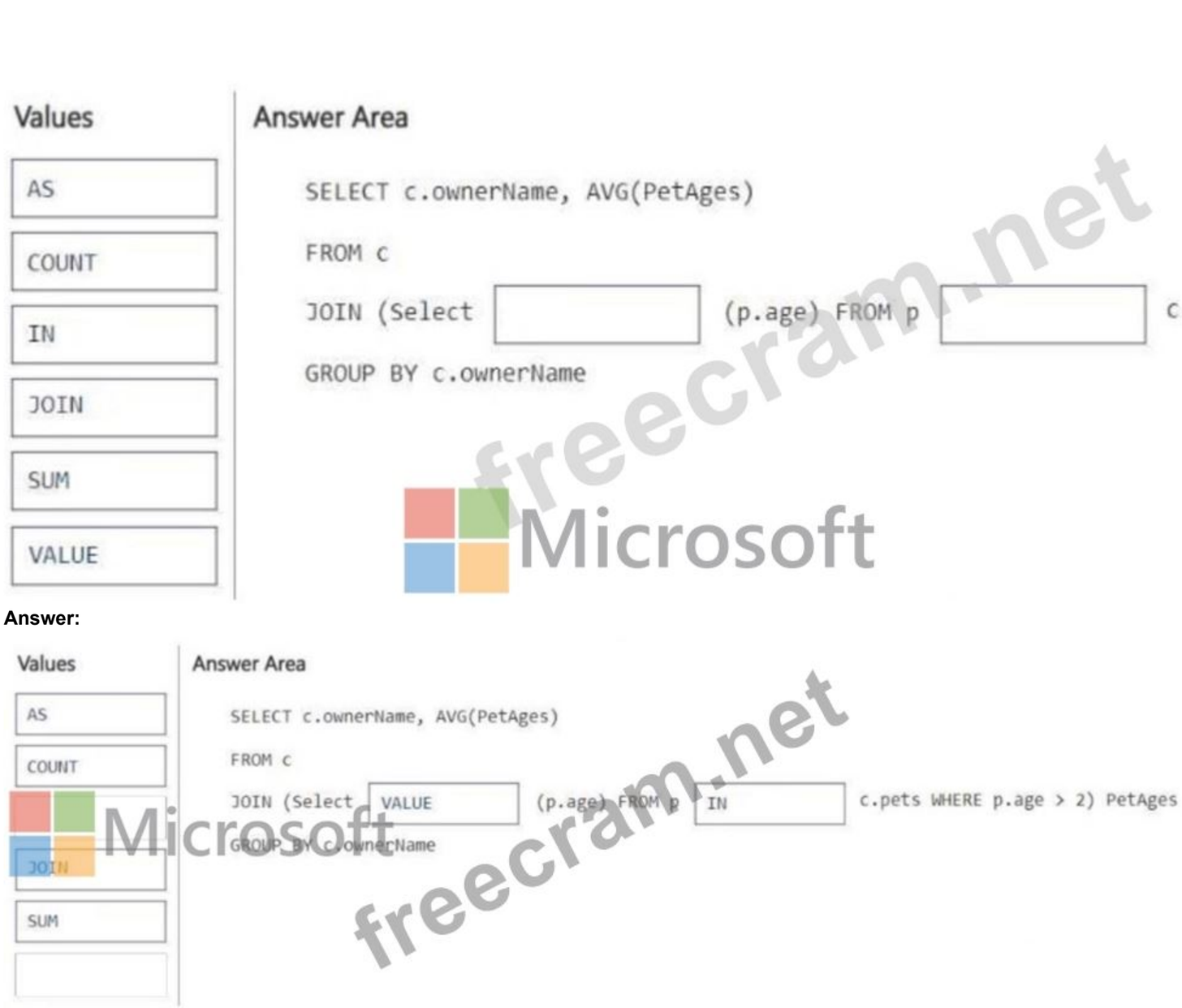
Answer Area

```
SELECT c.ownerName, AVG(PetAges)
FROM c
JOIN (Select [          ] (p.age) FROM p [          ] c.pets WHERE p.age > 2) PetAges
GROUP BY c.ownerName
```

freecram.net

Microsoft

Answer:



Microsoft

freecram.net

NEW QUESTION: 63

You have an Azure Cosmos DB for NoSQL container named Container1.

You run SQL queries against Container1 by using the Azure Cosmos DB .NET SDK.

You need to identify the number of request units consumed by each query.

Which two items can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. IDocumentQuery

B. RequestOptions

C. StoredProcedureResponse
D. ResourceResponse
Answer: (SHOW ANSWER)
To identify the number of request units (RUs) consumed by each query using the Azure Cosmos DB .NET SDK (specifically v2 API / DocumentClient syntax implied by your choices), you should use IDocumentQuery and FeedResponse.
ResourceResponse: In the legacy Azure Cosmos DB .NET SDK (v2), query and item operations return a ResourceResponse<T> object. This object contains a RequestCharge property that explicitly yields the number of RUs consumed by the database operation.
IDocumentQuery: When you perform asynchronous pagination or streaming of SQL query results in the v2 .NET SDK, the IDocumentQuery<T> interface is implemented. This interface exposes properties and response fields that catch execution metrics-including total request charges-as you iterate through your data feed.
Reference:
https://learn.microsoft.com/sv-se/azure/cosmos-db/find-request-unit-charge

NEW QUESTION: 64
You are designing an Azure Cosmos DB for NoSQL solution to store data from IoT devices.
Writes from the devices will occur every second. Data will be retained indefinitely.
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"
}
```

freecram.net
Microsoft

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. Create a new synthetic key that contains deviceId and deviceManufacturer.

C. Use deviceId as the partition key.
D. Use deviceManufacturer as the partition key.
Answer: (SHOW ANSWER)

NEW QUESTION: 65
Hotspot Question
You are creating a database in an Azure Cosmos DB for NoSQL 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	○	○

Microsoft
freeexam.net

Answer:

Answer Area

Microsoft

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	○	○

freecram.net

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: 66
You have an Azure Cosmos DB for NoSQL account that hosts a container named Container1.
You implement a change feed processor.
You discover that the processor is processing changes more slowly than the changes occur in Container1.
You need to resolve the issue.
What should you use?
A. a change feed estimator
B. an integrated cache
C. a diagnostics full-text query
D. a stored procedure
Answer: (SHOW ANSWER)
To handle change feed lag in Azure Cosmos DB, you can use the Change Feed Estimator to monitor the number of pending changes. This metric acts as a signal to scale your Change Feed Processor instances horizontally.
How the Estimator Works
The Estimator is a "sidecar" logic that measures the difference between the last processed item (stored in the lease container) and the latest change in the monitored container.
Frequency: It typically polls every 5 seconds (customizable).

Resource Usage: Each estimation consumes Request Units (RUs) from both the monitored and lease containers.
Independence: It can be deployed as a standalone instance separate from your processors to avoid impacting their performance.
Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/change-feed-processor

NEW QUESTION: 67
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 for NoSQL account named account1 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 application to use the change feed processor to read the change feed and you configure the application to trigger the function.
Does this meet the goal?
A. Yes
B. No
Answer: (SHOW ANSWER)
Instead configure an Azure Monitor alert to trigger the function. You can set up alerts from the Azure Cosmos DB pane or the Azure Monitor service in the Azure portal.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/create-alerts

NEW QUESTION: 68
Hotspot Question
You have an Azure subscription. The subscription contains an Azure Cosmos DB for NoSQL account named account1 that hosts a container named Customers. Multi-region writes are allowed.

```
var sales = await createClient.GetDatabase( "sales" );
var customers = await sales.CreateContainerIfNotExistsAsync
(
    new ContainerProperties("customers", "/id")
    {
        ConflictResolutionPolicy = new ConflictResolutionPolicy()
        {
            Mode = ConflictResolutionMode.Custom,
            ResolutionProcedure = "dbs/sales/colls/customers/sprocs/conflict"
        }
    }
);

await customers.Scripts.CreateStoredProcedureAsync
(
    new StoredProcedureProperties("conflict", File.ReadAllText(@"conflict.js"))
);
```

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
`conflict` is the ID of the stored procedure used for conflict resolution.	○	○
When a write conflict occurs, the `id` property of items in `customers` will hold the session token of the last writer.	○	○
Changing the conflict resolution policy for `customers` to `LastWriterWins` requires dropping the container.	○	○

Answer:

Microsoft
freecram.net

Answer Area

Statements	Yes	No
`conflict` is the ID of the stored procedure used for conflict resolution.	○	○
When a write conflict occurs, the `id` property of items in `customers` will hold the session token of the last writer.	○	○
Changing the conflict resolution policy for `customers` to `LastWriterWins` requires dropping the container.	○	○

Microsoft freecram.net

NEW QUESTION: 69

Hotspot Question

You have three containers in an Azure Cosmos DB Core (SQL) API account as shown in the following table.

Name	Database	Time to Live
cn1	db1	On (no default)
cn2	db1	Off
cn3	db1	On (no default)

You have the following Azure functions:

- A function named Fn1 that reads the change feed of cn1
- A function named Fn2 that reads the change feed of cn2
- A function named Fn3 that reads the change feed of cn3

You perform the following actions:

- Delete an item named item1 from cn1.
- Update an item named item2 in cn2.

For an item named item3 in cn3, update the item time to live to 3,600 seconds. For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Statements	Yes	No
Fn1 will receive item1 from the change feed	○	○
Fn2 can check the _etag of item2 to see whether the item is an update or an insert	○	○
Fn3 will receive item3 from the change feed	○	○

freecram.net
Microsoft

Answer:

Statements	Yes	No
Fn1 will receive item1 from the change feed	○	●
Fn2 can check the _etag of item2 to see whether the item is an update or an insert	○	●
Fn3 will receive item3 from the change feed	●	○

Microsoft
freecram.net

Explanation:

Box 1: No

Azure Cosmos DB's change feed is a great choice as a central data store in event sourcing architectures where all data ingestion is modeled as writes (no updates or deletes). Note: The change feed does not capture deletes. If you delete an item from your container, it is also removed from the change feed. The most common method of handling this is adding a soft marker on the items that are being deleted. You can add a property called "deleted" and set it to

"true" at the time of deletion. This document update will show up in the change feed. You can set a TTL on this item so that it can be automatically deleted later.

Box 2: No

The _etag format is internal and you should not take dependency on it, because it can change anytime.

Box 3: Yes

Change feed support in Azure Cosmos DB works by listening to an Azure Cosmos container for any changes.

Reference:

https://docs.microsoft.com/en-us/azure/cosmos-db/sql/change-feed-design-patterns

https://docs.microsoft.com/en-us/azure/cosmos-db/change-feed

NEW QUESTION: 70

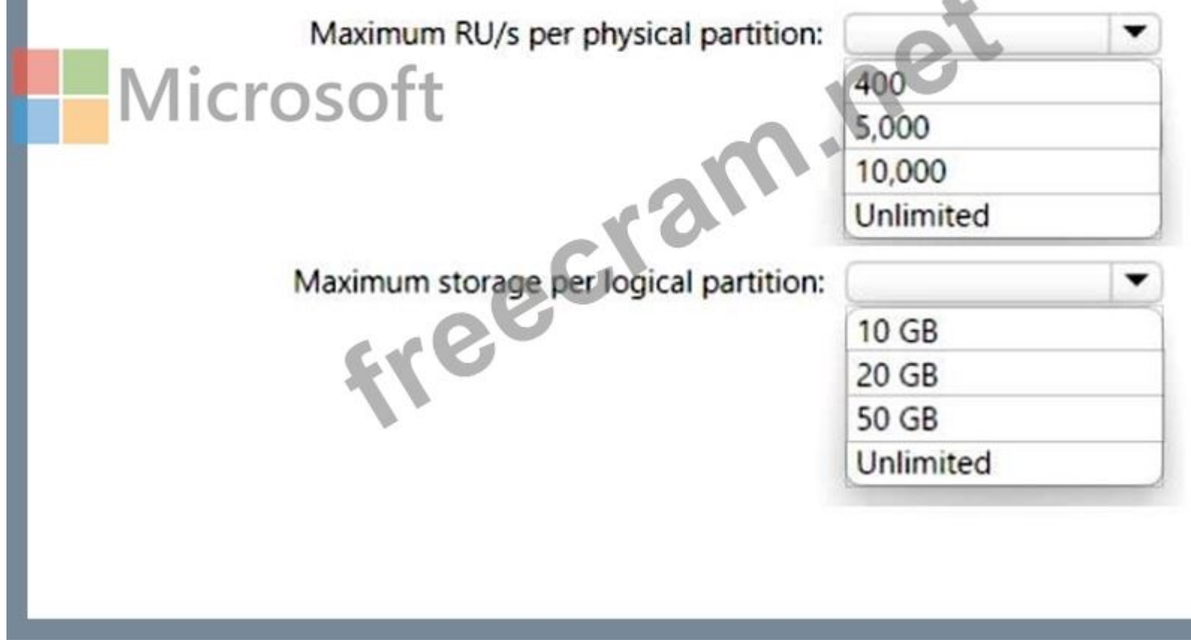
Hotspot Question

You are designing a data model for an Azure Cosmos DB for NoSQL account.

What are the partition limits for request units per second (RU/s) and storage? To answer, select the appropriate options in the answer area.

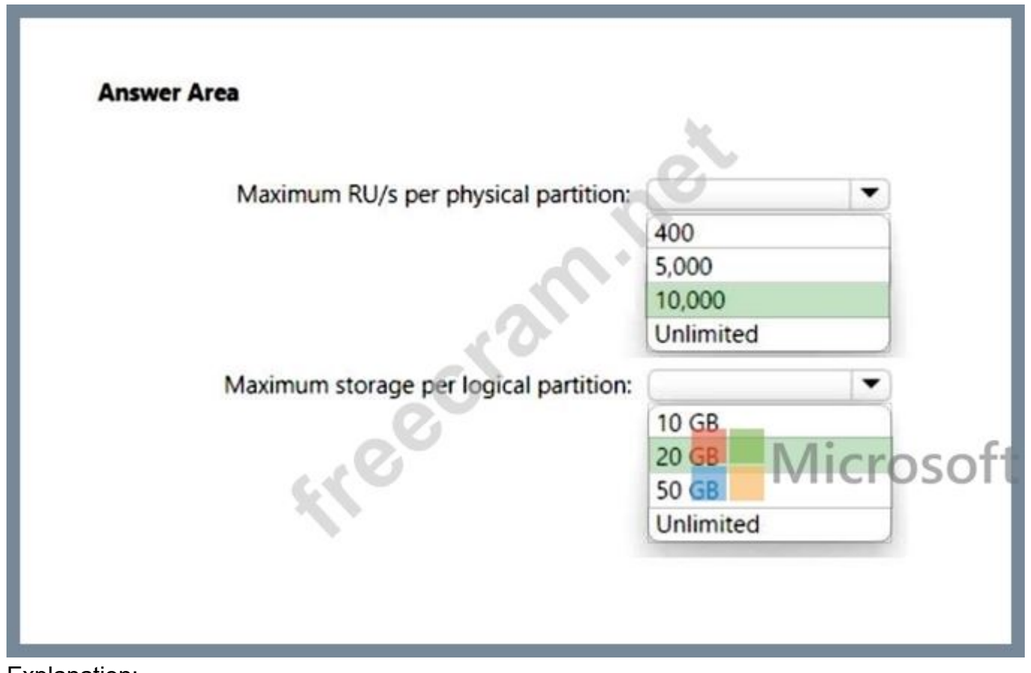
NOTE: Each correct selection is worth one point.

Answer Area



Microsoft
freecram.net

Answer:



Explanation:
https://learn.microsoft.com/en-us/azure/cosmos-db/concepts-limits

NEW QUESTION: 71
You are designing an Azure Cosmos DB for NoSQL solution to store data from IoT devices.
Writes from the devices will occur every second. Data will be retained indefinitely.
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"
}
```


freecram.net



Microsoft

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 deviceId as the partition key.
B. Create a new Synthetic key that contains deviceId and deviceManufacturer.
C. Use deviceManufacturer as the partition key.
D. Create a new synthetic key that contains deviceId and a random number.

Answer: D (LEAVE A REPLY)

Creating a synthetic key that combines deviceId with a random number helps distribute writes more evenly across partitions, effectively minimizing partition skew and avoiding hot partitions.
This approach also helps to maintain a balance of write operations, ensuring that no single partition is overwhelmed, which can occur if using deviceId alone, especially if devices write data in bursts.

NEW QUESTION: 72

Hotspot Question
You have an Azure Cosmos DB container named container1 that has a provisioned throughput and two physical partitions.
You monitor the following metrics for container1:
- Normalized RU consumption
- The percentage of requests that have an HTTP status code of 429

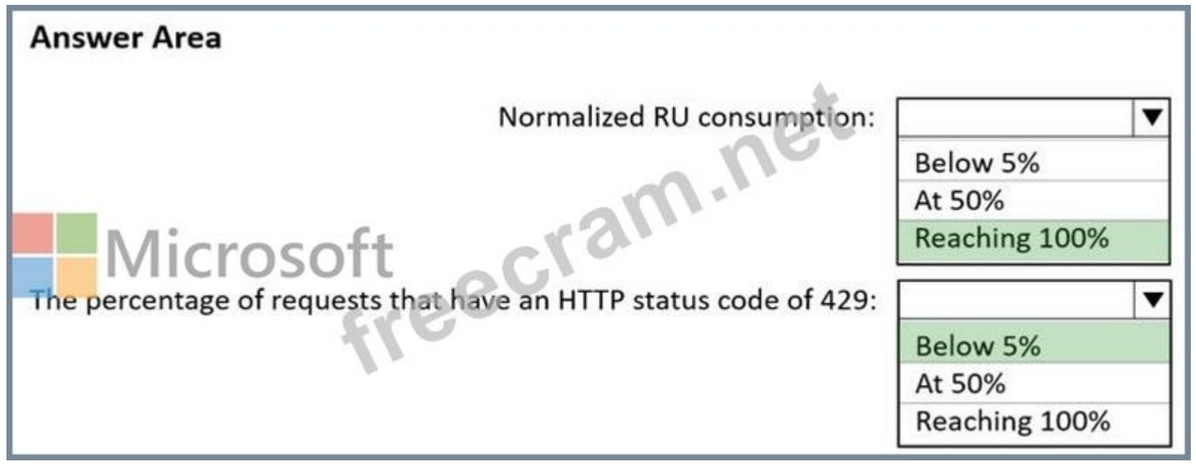
You need to confirm that contained is configured to maximize resource utilization.
What are the optimal values for each metric? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area



freecram.net
Microsoft

Answer:



Microsoft
freecram.net

NEW QUESTION: 73

You have an Azure Cosmos DB for NoSQL account that has multiple write regions.

You need to receive an alert when requests that target the database exceed the available request units per second (RU/s).

Which Azure Monitor signal should you use?

A. Metadata Requests

B. Document Count

C. Data Usage

D. Document Quota

Answer: (SHOW ANSWER)

For this question there are the correct answers:

* Total Request Units

Used to monitor Total RU usage at a minute granularity. To get average RU consumed per second, use Sum aggregation at minute interval/level and divide by 60.

* Metadata Requests

Count of metadata requests. Azure Cosmos DB maintains system metadata container for each account, that allows you to enumerate collections, databases, etc., and their configurations, free of charge.

Used to monitor metadata requests in scenarios where requests are being throttled.

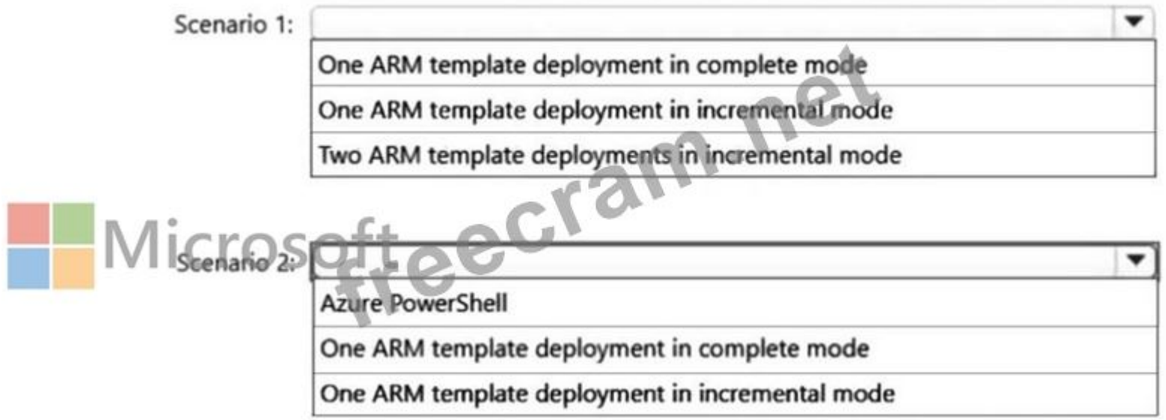
For this question there are the incorrect answers:
* Data Usage
Used to monitor total data usage at container and region, minimum granularity should be 5 minutes.
* Document Count
Used to monitor document count at container and region, minimum granularity should be 5 minutes.
* Document Quota
Used to monitor total quota at container and region, minimum granularity should be 5 minutes.
* Provisioned Throughput
Used to monitor provisioned throughput per container.
* Region Added
* Region Removed
Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/monitor-reference

NEW QUESTION: 74
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
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: (SHOW ANSWER)
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

NEW QUESTION: 75
Hotspot Question
You have an Azure Resource Manager (ARM) template that provisions Azure Cosmos DB accounts and databases. The accounts are deployed to a single Azure region. The databases have provisioned throughput configured. The name of the region, the type of database throughput, and the throughput value are defined as template parameters.
You need to modify some of the Azure Cosmos DB accounts and databases based on the following scenarios:
- Scenario 1: Add a read region and increase the database throughput.
- Scenario 2: Change the provisioned database throughput to autoscale.
The solution must minimize administrative effort and the duration of the modifications.
What should you use for each scenario? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

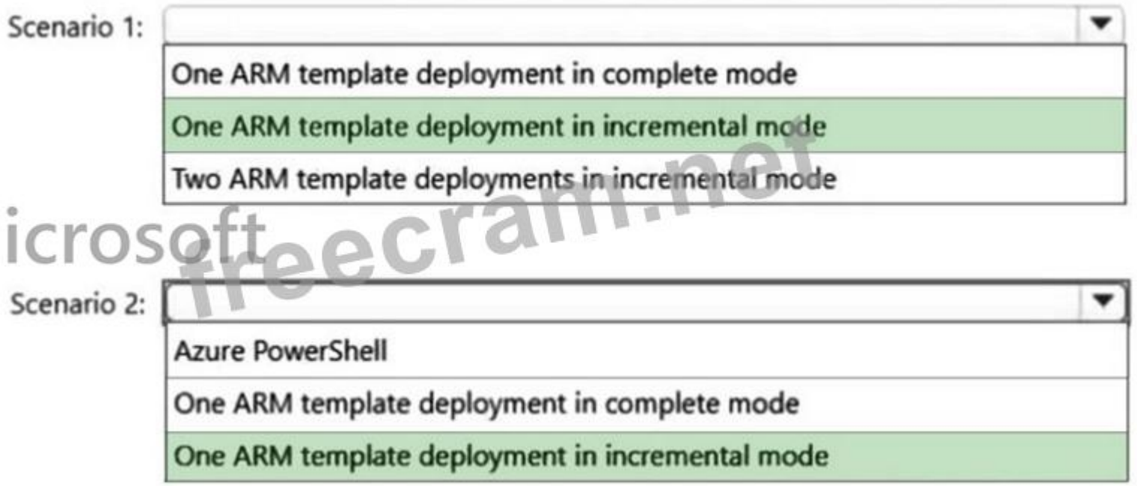
Answer Area



Microsoft freecram.net

Answer:

Answer Area



Microsoft freecram.net

NEW QUESTION: 76

Hotspot Question

You have an Azure Cosmos DB for NoSQL account.

You plan to create a container named container1. The container1 container will store items that include two properties named name and age.

The most commonly executed queries will query container1 for a specific name. The following is a sample of the query.

```
SELECT*
FROM c
WHERE c.name = 'ben smith'
ORDER BY c.name DESC, c.age ASC
```

You need to define an opt-in indexing policy for container1. The solution must meet the following requirements:
- Minimize the number of request units consumed by the queries.
- Ensure that the _etag property is excluded from indexing.
How should you define the indexing policy? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
{
    "automatic": true,
    "indexingMode": "Consistent",
    "includePaths": [],
    "excludePaths": [ ▼ ]
        []
        [{"path": "/*"}],
        [{"path": "/\"_etag\"/?"}],
    "compositeIndexes":[
        [
            {
                "path": [ ▼ ]
                    "/name",
                    "/name/*",
                    "/name/?",
                "order": "ascending"
            },
            {
                "path": "/age",
                "order": "descending"
            }
        ]
    ]
}
```

Answer:

freecram.net
Microsoft

Answer Area

```
{
    "automatic": true,
    "indexingMode": "Consistent",
    "includePaths": [],
    "excludePaths":
                    []
                    [{"path": "/*"}],
                    [{"path": "/\"_etag\"/?"}],
    "compositeIndexes":[
            [
                {
                    "path":
                                "/name",
                                "/name/*",
                                "/name/?",
                    "order": "ascending"
                },
                {
                    "path": "/age",
                    "order": "descending"
                }
            ]
        ]
    }
```

freecram.net

Valid DP-420 Dumps shared by EduDump.com for Helping Passing DP-420 Exam! EduDump.com now offer the **newest DP-420 exam dumps**, the EduDump.com DP-420 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com DP-420 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/DP-420/premium/ (**281** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 77
You have an Azure Cosmos DB for NoSQL 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: A (LEAVE A REPLY)
All conflicts related data is stored in the conflicts feed, you have to orchestrate the function using a timer.

NEW QUESTION: 78
You have the following query.

```
SELECT * FROM c
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: (SHOW ANSWER)
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/

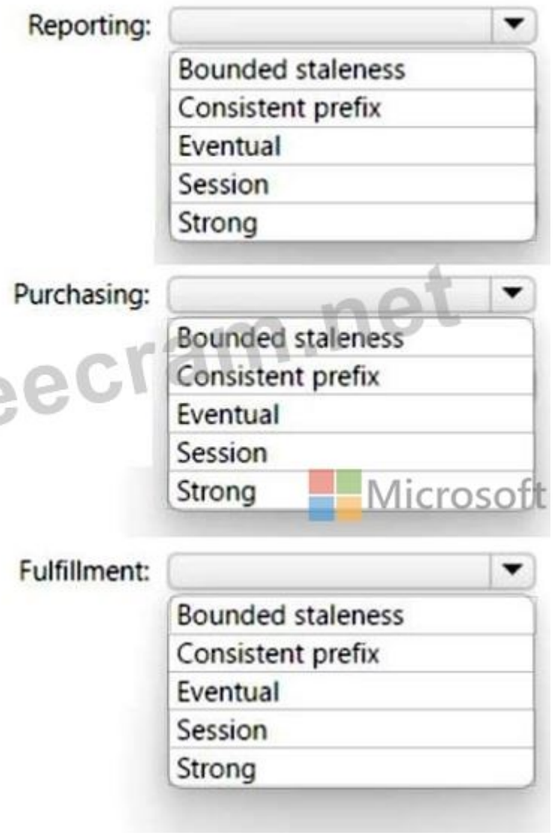
NEW QUESTION: 79
Hotspot Question
You plan to use a multi-region Azure Cosmos DB for NoSQL account to store data for a new application suite. The suite contains the applications shown in the following table.

Name	Requirement
Reporting	Must be able to track the total order counts within five minutes of orders being placed.
Purchasing	Must guarantee that the latest committed stock quantities are used always.
Fulfillment	Must ensure that orders are read in the order in which they are placed.

Each application should use the weakest consistency level possible.
Which consistency level should you configure for each application? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

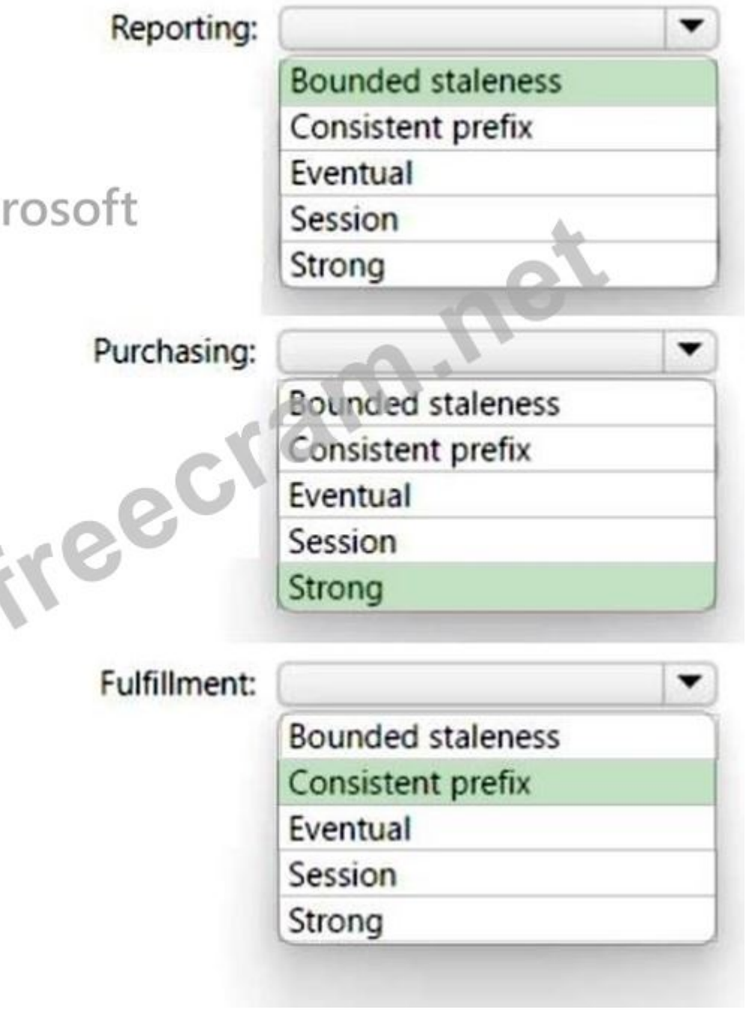
Answer Area



freecram.net
Microsoft

Answer:

Answer Area



Microsoft
freecram.net

Explanation:
https://willvelida.medium.com/understanding-consistency-levels-in-azure-cosmos-db-77b90dbf3daa

NEW QUESTION: 80
Hotspot Question
You have a container named container1 in an Azure Cosmos DB for NoSQL account. The container1 container has 120 GB of data.
The following is a sample of a document in container1.

```
{
    "customerId" : "5425",
    "orderId" : "9d7816e6-f401-42ba-ad05-0e03de35c0b8",
    "orderDate" : "2019-05-03",
    "orderDetails" : []
}
```

The orderId property is used as the partition key.

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

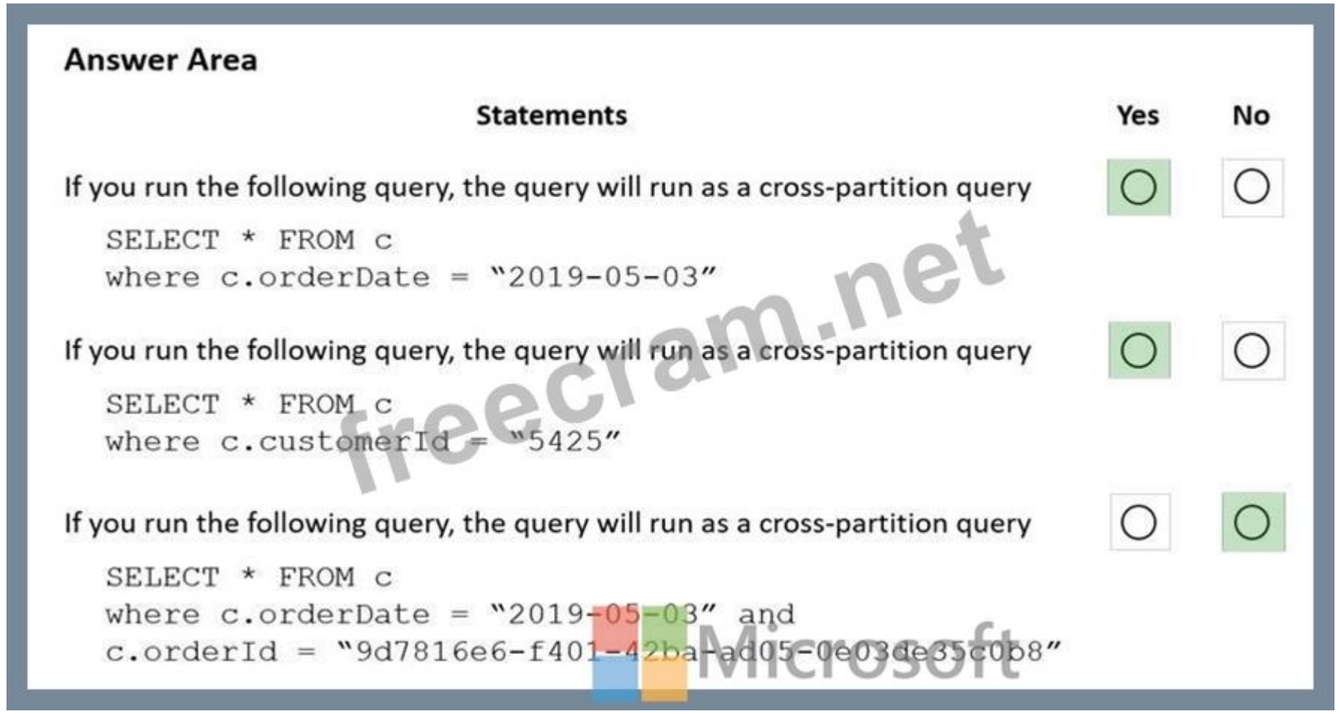
If you run the following query, the query will run as a cross-partition query
`SELECT * FROM c`
`where c.orderDate = "2019-05-03"`
	○	○

If you run the following query, the query will run as a cross-partition query
`SELECT * FROM c`
`where c.customerId = "5425"`
	○	○

If you run the following query, the query will run as a cross-partition query
`SELECT * FROM c`
`where c.orderDate = "2019-05-03" and`
`c.orderId = "9d7816e6-f401-42ba-ad05-0e03de35c0b8"`
	○	○

Answer:

freecram.net



Explanation:

Box 1: Yes

Records with different OrderIDs will match.

Box 2: Yes

Records with different OrderIDs will match.

Box 3: No

Only records with one specific OrderId will match

NEW QUESTION: 81

Drag and Drop Question

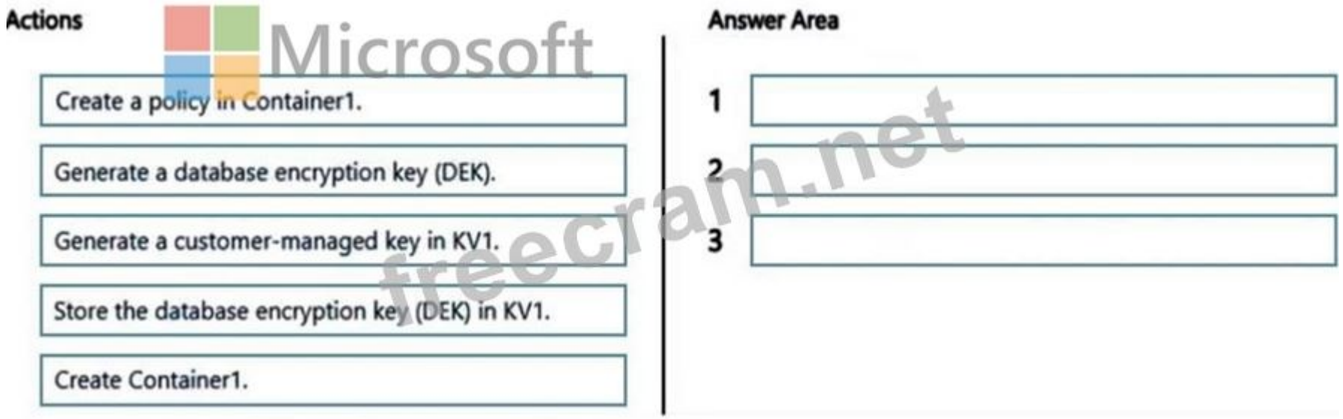
You have an Azure subscription that contains the resources shown in the following table.

Name	Description
CosmosDB1	Azure Cosmos DB for NoSQL database
KV1	Azure key vault configured with access policies

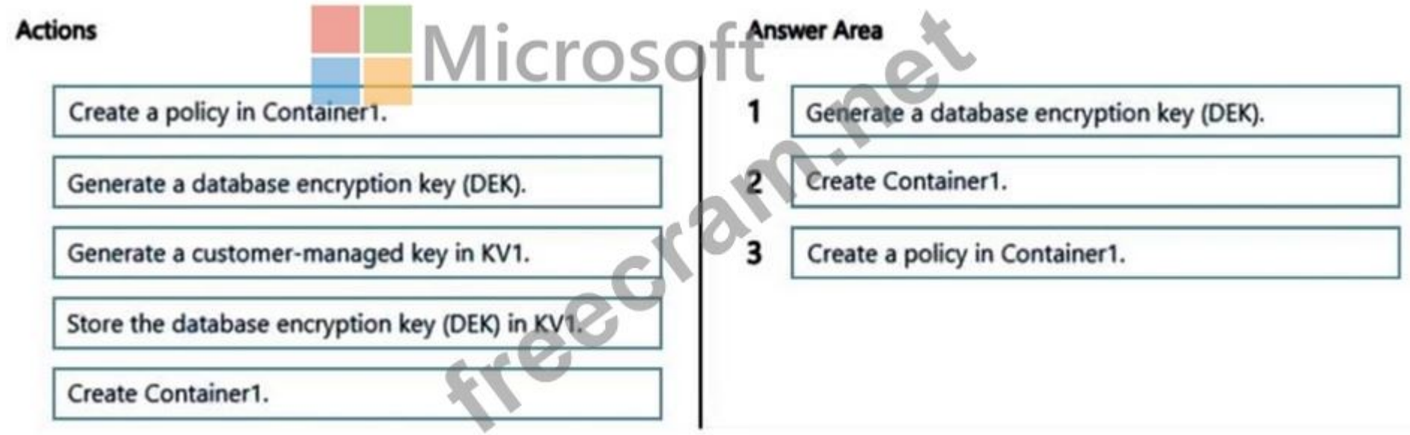
In CosmosDB1, you plan to create a container named Container1 that will store sensitive data.

You need to ensure that Container1 uses Always Encrypted for Azure Cosmos DB.

Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.



Answer:



NEW QUESTION: 82

Case Study 1 - Litware, inc

Overview

Litware, Inc. is a United States-based grocery retailer. Litware has a main office and a primary datacenter in Seattle. The company has 50 retail stores across the United States and an emerging online presence. Each store connects directly to the internet. Existing environment.

Cloud and Data Service Environments. Litware has an Azure subscription that contains the resources shown in the following table.

Name	Type	Description
account1	Azure Cosmos DB for NoSQL account	The account has a single read-write region and contains a database named productdb. The productdb database contains two containers named con-product and con-productVendor. The account uses the session default consistency level.
iothub1	Azure IoT hub	The IoT hub collects per-second temperature and humidity telemetry from 5,000 IoT devices in the retail stores. A single telemetry reading generates 1 KB of data.
streamanalytics1	Azure Stream Analytics job	The job processes telemetry data collected from iothub1.

Each container in productdb is configured for manual throughput. The con-product container stores the company's product catalog data. Each document in con-product includes a con- productvendor value. Most queries targeting the data in con-product are in the following format.

SELECT * FROM con-product p WHERE p.con-productVendor - 'name'

Most queries targeting the data in the con-productVendor container are in the following format SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Existing environment.

Current Problems.

Litware identifies the following issues:

- Updates to product categories in the con-productVendor container do not propagate automatically to documents in the con-product container.
- Application updates in con-product frequently cause HTTP status code 429 "Too many requests".
- You discover that the 429 status code relates to excessive request unit (RU) consumption during the updates.

Requirements.

Planned Changes

Litware plans to implement a new Azure Cosmos DB for NoSQL account named account2 that will contain a database named iotdb. The iotdb database will contain two containers named con- iot1 and con-iot2.

Litware plans to make the following changes:

- Store the telemetry data in account2.
- Configure account1 to support multiple read-write regions.
- Implement referential integrity for the con-product container.
- Use Azure Functions to send notifications about product updates to different recipients.
- Develop an app named App1 that will run from all locations and query the data in account1.
- Develop an app named App2 that will run from the retail stores and query the data in account2.
- App2 must be limited to a single DNS endpoint when accessing account2.

Requirements.

Business Requirements

Litware identifies the following business requirements:

- 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.
- Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.
- Minimize the number of firewall changes in the retail stores.

Requirements.

Product Catalog Requirements

Litware identifies the following requirements for the product catalog:

- Implement a custom conflict resolution policy for the product catalog data.
- Minimize the frequency of errors during updates of the con-product container.
- Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.

- Trigger the execution of two Azure functions following every update to any document in the con- product container.
You are troubleshooting the current issues caused by the application updates.
Which action can address the application updates issue without affecting the functionality of the application?
A. Enable time to live for the con-product container.
B. Set the default consistency level of account1 to strong.
C. Set the default consistency level of account1 to bounded staleness.
D. Add a custom indexing policy to the con-product container.
Answer: (SHOW ANSWER)
For 429's on updating documents it is recommended to create a custom index policy, and only contain the properties that are needed.
https://learn.microsoft.com/en-us/azure/cosmos-db/sql/troubleshoot-request-rate-too-large?tabs=resource-specific#429s-on-create-replace-or-upsert-document-requests

NEW QUESTION: 83
Case Study 1 - Litware, inc
Overview
Litware, Inc. is a United States-based grocery retailer. Litware has a main office and a primary datacenter in Seattle. The company has 50 retail stores across the United States and an emerging online presence. Each store connects directly to the internet. Existing environment.
Cloud and Data Service Environments. Litware has an Azure subscription that contains the resources shown in the following table.

Name	Type	Description
account1	Azure Cosmos DB for NoSQL account	The account has a single read-write region and contains a database named productdb. The productdb database contains two containers named con-product and con-productVendor. The account uses the session default consistency level.
iothub1	Azure IoT hub	The IoT hub collects per-second temperature and humidity telemetry from 5,000 IoT devices in the retail stores. A single telemetry reading generates 1 KB of data.
streamanalytics1	Azure Stream Analytics job	The job processes telemetry data collected from iothub1.

Each container in productdb is configured for manual throughput. The con-product container stores the company's product catalog data. Each document in con-product includes a con- productvendor value. Most queries targeting the data in con-product are in the following format.
SELECT * FROM con-product p WHERE p.con-productVendor - 'name'
Most queries targeting the data in the con-productVendor container are in the following format SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Existing environment.
Current Problems.
Litware identifies the following issues:
- Updates to product categories in the con-productVendor container do not propagate automatically to documents in the con-product container.
- Application updates in con-product frequently cause HTTP status code 429 "Too many requests".
- You discover that the 429 status code relates to excessive request unit (RU) consumption during the updates.
Requirements.
Planned Changes
Litware plans to implement a new Azure Cosmos DB for NoSQL account named account2 that will contain a database named iotdb. The iotdb database will contain two containers named con- iot1 and con-iot2.
Litware plans to make the following changes:
- Store the telemetry data in account2.
- Configure account1 to support multiple read-write regions.

- Implement referential integrity for the con-product container.
- Use Azure Functions to send notifications about product updates to different recipients.
- Develop an app named App1 that will run from all locations and query the data in account1.
- Develop an app named App2 that will run from the retail stores and query the data in account2.
- App2 must be limited to a single DNS endpoint when accessing account2.

Requirements.

Business Requirements

Litware identifies the following business requirements:

- 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.
- Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.
- Minimize the number of firewall changes in the retail stores.

Requirements.

Product Catalog Requirements

Litware identifies the following requirements for the product catalog:

- Implement a custom conflict resolution policy for the product catalog data.
- Minimize the frequency of errors during updates of the con-product container.
- Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.
- Trigger the execution of two Azure functions following every update to any document in the con- product container.

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 (LEAVE A REPLY)

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: 84

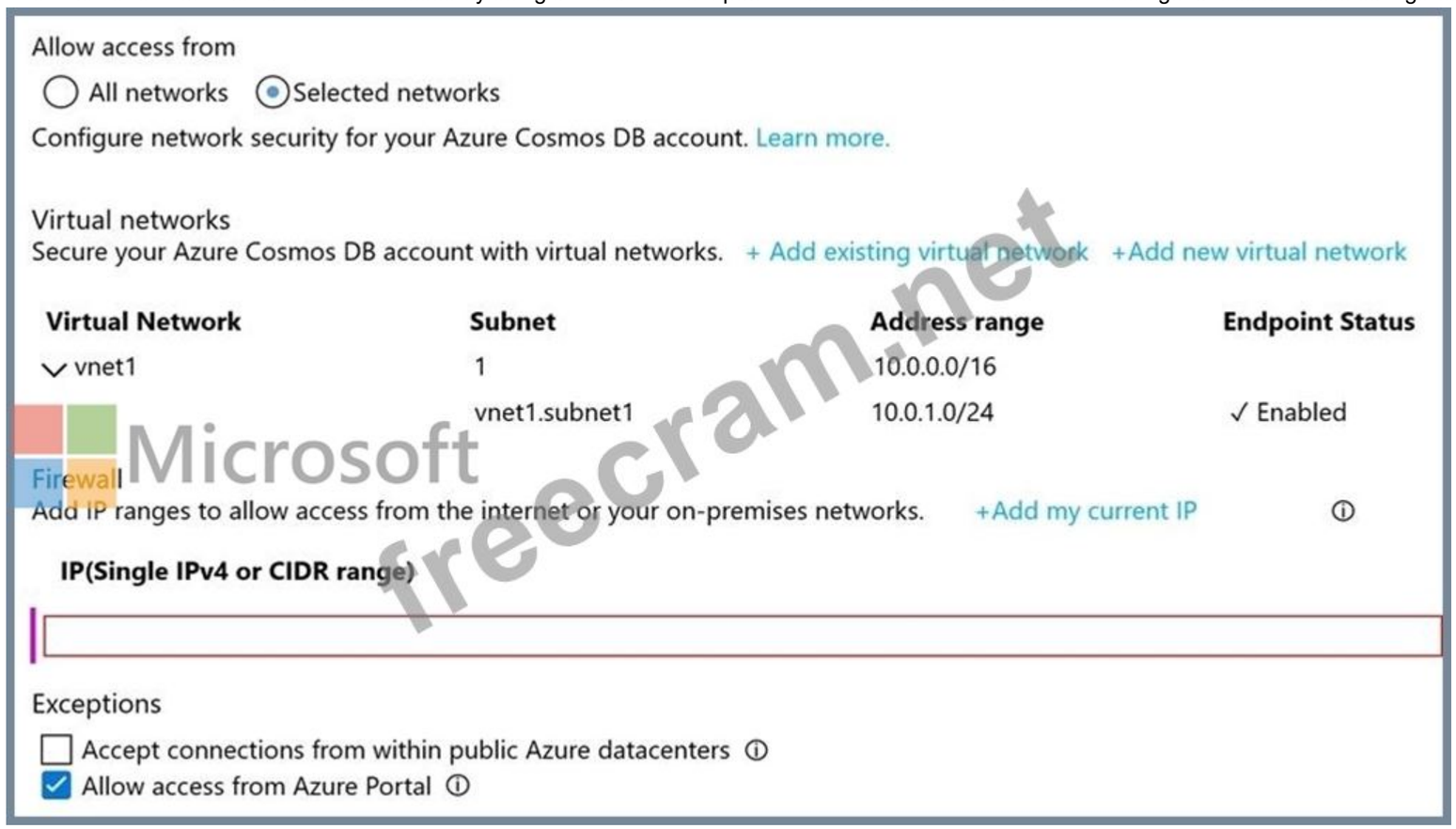
Hotspot Question

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.

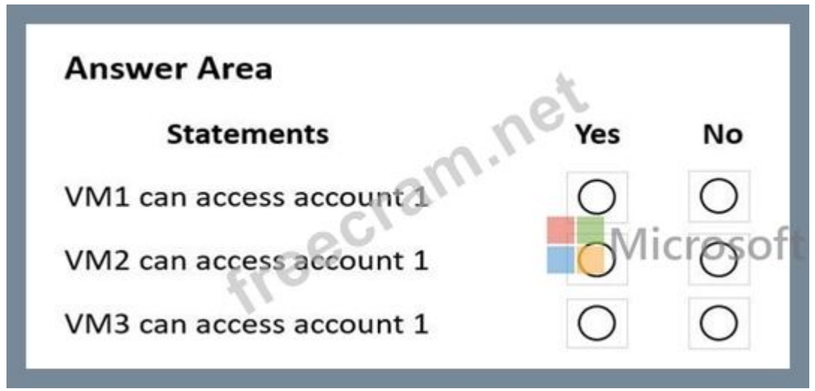
Subnet	Network	IP address range	Virtual machine
subnet1	vnet1	10.0.0.0/24	VM1
subnet2	vnet1	10.0.1.0/24	VM2
subnet3	vnet2	10.1.0.0/24	VM3

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.



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:



Explanation:

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

NEW QUESTION: 85

Hotspot Question

You have a database in an Azure Cosmos DB for NoSQL account that is used for development.

The database is modified once per day in a batch process.

You need to ensure that you can restore the database if the last batch process fails. The solution must minimize costs.

How should you configure the backup settings? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area



freecram.net
Microsoft

Answer:

Answer Area



freecram.net
Microsoft

NEW QUESTION: 86

You have a global ecommerce application that stores data in an Azure Cosmos DB for NoSQL account. The account is configured for multi-region writes.

You need to create a stored procedure for a custom conflict resolution policy for a new container.

In the event of a conflict caused by a deletion, the deletion must always take priority.

Which parameter should you check in the stored procedure function?

A. isTombstone

B. conflictingItems
C. existingItem
D. incomingItem
Answer: (SHOW ANSWER)
isTombstone: Boolean indicating if the incomingItem is conflicting with a previously deleted item.
When true, existingItem is also null.
Incorrect:
conflictingItems: Array of the committed version of all items in the container that are conflicting with incomingItem on ID or any other unique index properties. existingItem: The currently committed item. This value is non-null in an update and null for an insert or deletes.
incomingItem: The item being inserted or updated in the commit that is generating the conflicts. Is null for delete operations.
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-manage-conflicts

NEW QUESTION: 87
You have an Azure Cosmos DB for NoSQL account that contains a container named customers.
You have an application named App1 that caches customers and must be notified when the list of customers changes.
You implement a change feed. The Time to Live settings have the default values.
You discover the following:
- Notifications for INSERT and UPDATE operations are received within 10 seconds.
- Notifications for DELETE operations are NOT received.
Which two actions should you perform to ensure that App1 receives notifications for DELETE operations? Each correct answer present part of the solution.
NOTE: Each correct selection is worth one point.
A. Add a post-trigger.
B. Use an UPDATE instead of a DELETE.
C. Add an IsDeleted flag to the items.
D. Increase the TTL.
E. Add a pre-trigger.
Answer: B,C (LEAVE A REPLY)
Implement "Soft Deletes"
You can use a soft-delete pattern to make deletions visible to the standard change feed.
The Process:
Update, don't delete: Instead of a hard delete, update the contact with a property like "isDeleted": true.
Set TTL: Apply a small Time to Live (TTL) value (e.g., 60 seconds) to that specific item.
Capture Change: The change feed sees this as an update and notifies your app.
Final Deletion: Azure Cosmos DB will automatically hard-delete the item once the TTL expires.
Reference:
https://docs.azure.cn/en-us/cosmos-db/change-feed-design-patterns

NEW QUESTION: 88
Hotspot Question

You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account. The account hosts a container named Container1 that stores product inventory data in the following format.

```
{
  "name": "Snow coat",
  "inventory": [
    {
      "location": "Redmond, WA",
   "quantity": 50
  },
  {
   "location": "Seattle, WA",
   "quantity": 30
  },
  {
   "location": "Boston, MA",
   "quantity": 25
  }
  ]
}
```

freecram.net

Microsoft

You need to query the average inventory quantity of each product in Washington, including the product name.
How should you complete the query? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
SELECT
  p.name,
  [inventoryData.inventoryAverage ▼]
FROM
  products p
  [WHERE ▼]
  (SELECT
    AVG(i.quantity) AS inventoryAverage
  FROM i IN p.inventory
  WHERE ENDSWITH(i.location, "WA")) AS inventoryData
```

- inventoryData.inventoryAverage
  - i.inventoryAverage
  - inventoryData.inventoryAverage
  - p.inventoryAverage
- WHERE
  - INNER JOIN
  - JOIN
  - WHERE

Microsoft
freecram.net

Answer:

Answer Area

```
SELECT
  p.name,
  [inventoryData.inventoryAverage ▼]
FROM
  products p
  [WHERE ▼]
  (SELECT
    AVG(i.quantity) AS inventoryAverage
  FROM i IN p.inventory
  WHERE ENDSWITH(i.location, "WA")) AS inventoryData
```

- inventoryData.inventoryAverage
  - i.inventoryAverage
  - inventoryData.inventoryAverage
  - p.inventoryAverage
- WHERE
  - INNER JOIN
  - JOIN
  - WHERE

Microsoft
freecram.net

NEW QUESTION: 89
You have an Azure Cosmos DB for NoSQL account configured for global distribution across four regions.
At connection time, the SQL SDK needs to identify the optimal endpoint for reading and writing.
Which two factors can influence the SDK? Each correct answer presents a complete solution.
NOTE: Each correct selection is worth one point.
A. the consistency level in RequestOptions
B. regional latency
C. the default consistency level
D. the PreferredLocations configuration
E. a region being available
Answer: D,E (LEAVE A REPLY)
In order to take advantage of global distribution, client applications can specify the ordered preference list of regions to be used to perform document operations. Based on the Azure Cosmos DB account configuration, current regional availability and the preference list specified, the most optimal endpoint will be chosen by the SQL SDK to perform write and read operations.
https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/tutorial-global-distribution?tabs=dotnetv2%2Capi-async#preferred-locations

NEW QUESTION: 90
You have an Azure Cosmos DB database named database1 that contains a container named container1. The container1 container stores product data and has the following indexing policy.
Which path will be indexed?



A. /product/brand/tailspin
B. /product/[]/category
C. /product/category
D. /product/brand
Answer: (SHOW ANSWER)

NEW QUESTION: 91
Case Study 1 - Litware, inc
Overview
Litware, Inc. is a United States-based grocery retailer. Litware has a main office and a primary datacenter in Seattle. The company has 50 retail stores across the United States and an emerging online presence. Each store connects directly to the internet. Existing environment.
Cloud and Data Service Environments. Litware has an Azure subscription that contains the resources shown in the following table.

Name	Type	Description
account1	Azure Cosmos DB for NoSQL account	The account has a single read-write region and contains a database named productdb. The productdb database contains two containers named con-product and con-productVendor. The account uses the session default consistency level.
iothub1	Azure IoT hub	The IoT hub collects per-second temperature and humidity telemetry from 5,000 IoT devices in the retail stores. A single telemetry reading generates 1 KB of data.
streamanalytics1	Azure Stream Analytics job	The job processes telemetry data collected from iothub1.

Each container in productdb is configured for manual throughput. The con-product container stores the company's product catalog data. Each document in con-product includes a con- productvendor value. Most queries targeting the data in con-product are in the following format.
SELECT * FROM con-product p WHERE p.con-productVendor - 'name'

Most queries targeting the data in the con-productVendor container are in the following format SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Existing environment.
Current Problems.
Litware identifies the following issues:
- Updates to product categories in the con-productVendor container do not propagate automatically to documents in the con-product container.
- Application updates in con-product frequently cause HTTP status code 429 "Too many requests".
- You discover that the 429 status code relates to excessive request unit (RU) consumption during the updates.
Requirements.
Planned Changes
Litware plans to implement a new Azure Cosmos DB for NoSQL account named account2 that will contain a database named iotdb. The iotdb database will contain two containers named con- iot1 and con-iot2.
Litware plans to make the following changes:
- Store the telemetry data in account2.
- Configure account1 to support multiple read-write regions.
- Implement referential integrity for the con-product container.
- Use Azure Functions to send notifications about product updates to different recipients.
- Develop an app named App1 that will run from all locations and query the data in account1.
- Develop an app named App2 that will run from the retail stores and query the data in account2.
- App2 must be limited to a single DNS endpoint when accessing account2.
Requirements.
Business Requirements
Litware identifies the following business requirements:
- 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.
- Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.
- Minimize the number of firewall changes in the retail stores.
Requirements.
Product Catalog Requirements
Litware identifies the following requirements for the product catalog:
- Implement a custom conflict resolution policy for the product catalog data.
- Minimize the frequency of errors during updates of the con-product container.
- Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.
- Trigger the execution of two Azure functions following every update to any document in the con- product container.
You need to provide a solution for the Azure Functions notifications following updates to con- product. The solution must meet the business requirements and the product catalog requirements.
Which two actions should you perform? Each correct answer presents part of the solution.
NOTE: Each correct selection is worth one point.
A. Configure the trigger for each function to use a different leaseCollectionPrefix
B. Configure the trigger for each function to use the same leaseCollectionNair.e
C. Configure the trigger for each function to use a different leaseCollectionName
D. Configure the trigger for each function to use the same leaseCollectionPrefix
Answer: (SHOW ANSWER)
leaseCollectionPrefix: when set, the value is added as a prefix to the leases created in the Lease collection for this Function. Using a prefix allows two separate Azure Functions to share the same Lease collection by using different prefixes.
Scenario: Use Azure Functions to send notifications about product updates to different recipients.

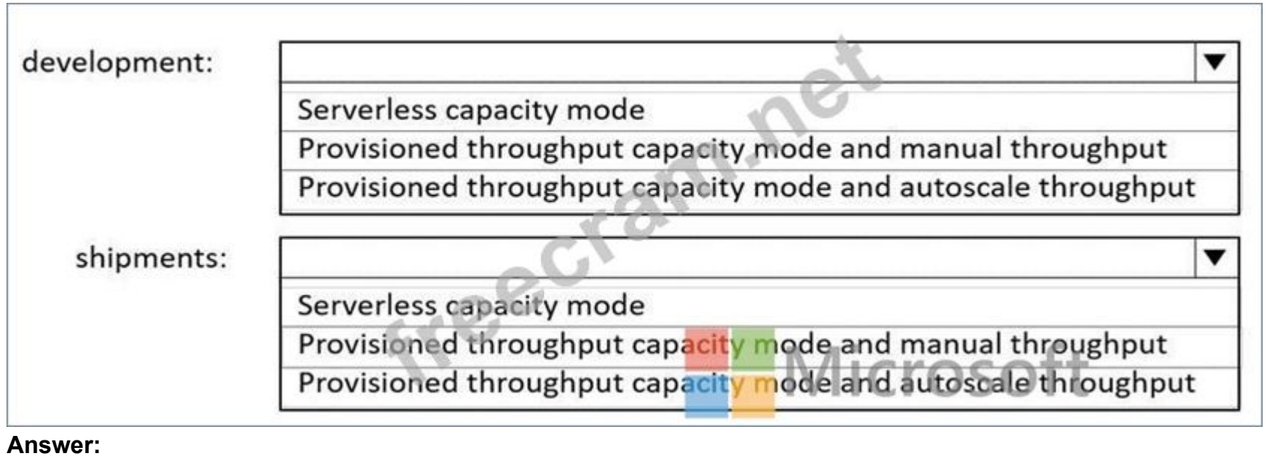
Trigger the execution of two Azure functions following every update to any document in the con- product container.
Reference:
https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-cosmosdb-v2-trigger

Valid DP-420 Dumps shared by EduDump.com for Helping Passing DP-420 Exam! EduDump.com now offer the **newest DP-420 exam dumps**, the EduDump.com DP-420 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com DP-420 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/DP-420/premium/ (**281** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

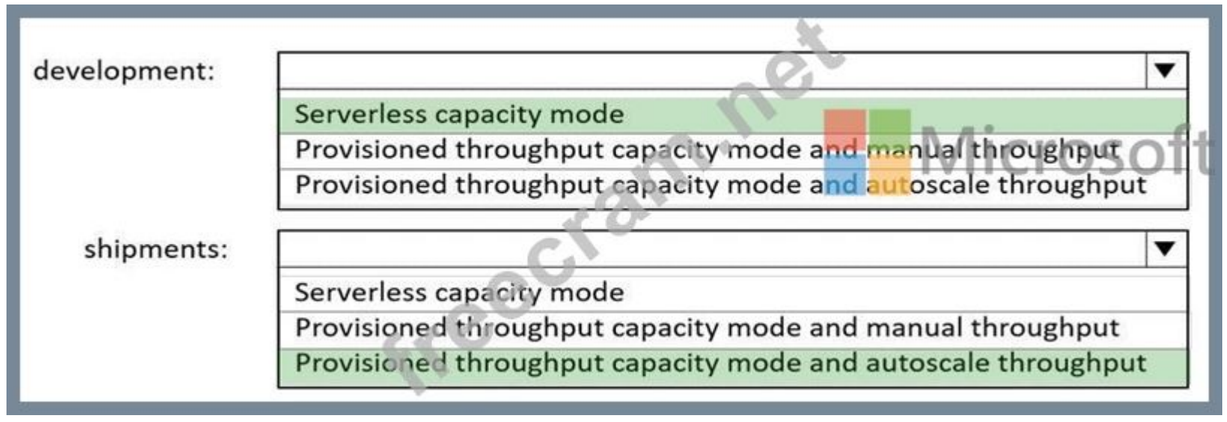
NEW QUESTION: 92
Hotspot Question
You plan to deploy two Azure Cosmos DB Core (SQL) API accounts that will each contain a single database. The accounts will be configured as shown in the following table.

Name	Description
development	• Supports the development of new application features • Used intermittently as needed during development
shipments	• Captures over 100,000 updates per second generated at unpredictable times throughout the business day • Used with Azure Synapse Link for analytics

How should you provision the containers within each account to minimize costs? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:



Explanation:
Box 1: Serverless capacity mode
Azure Cosmos DB serverless best fits scenarios where you expect intermittent and unpredictable traffic with long idle times. Because provisioning capacity in such situations isn't required and may be cost-prohibitive, Azure Cosmos DB serverless should be considered in the following use- cases:
Getting started with Azure Cosmos DB
Running applications with bursty, intermittent traffic that is hard to forecast, or low (<10%) average-to-peak traffic ratio Developing, testing, prototyping and running in production new applications where the traffic pattern is unknown Integrating with serverless compute services like Azure Functions Box 2: Provisioned throughput capacity mode and autoscale throughput The use cases of autoscale include:
Variable or unpredictable workloads: When your workloads have variable or unpredictable spikes in usage, autoscale helps by automatically scaling up and down based on usage. Examples include retail websites that have different traffic patterns depending on seasonality; IOT workloads that have spikes at various times during the day; line of business applications that see peak usage a few times a month or year, and more. With autoscale, you no longer need to manually provision for peak or average capacity.
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/serverless
https://docs.microsoft.com/en-us/azure/cosmos-db/provision-throughput-autoscale#use-cases-of-autoscale

NEW QUESTION: 93
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 could cause storage hot partitions
B. All the above
C. This partition key will cause fan out when filtering by countryId
D. This partition key will prevent throughput hot partitions
Answer: (SHOW ANSWER)

NEW QUESTION: 94
Hotspot Question
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	○	○

Microsoft
freecram.net

Answer:

Answer Area

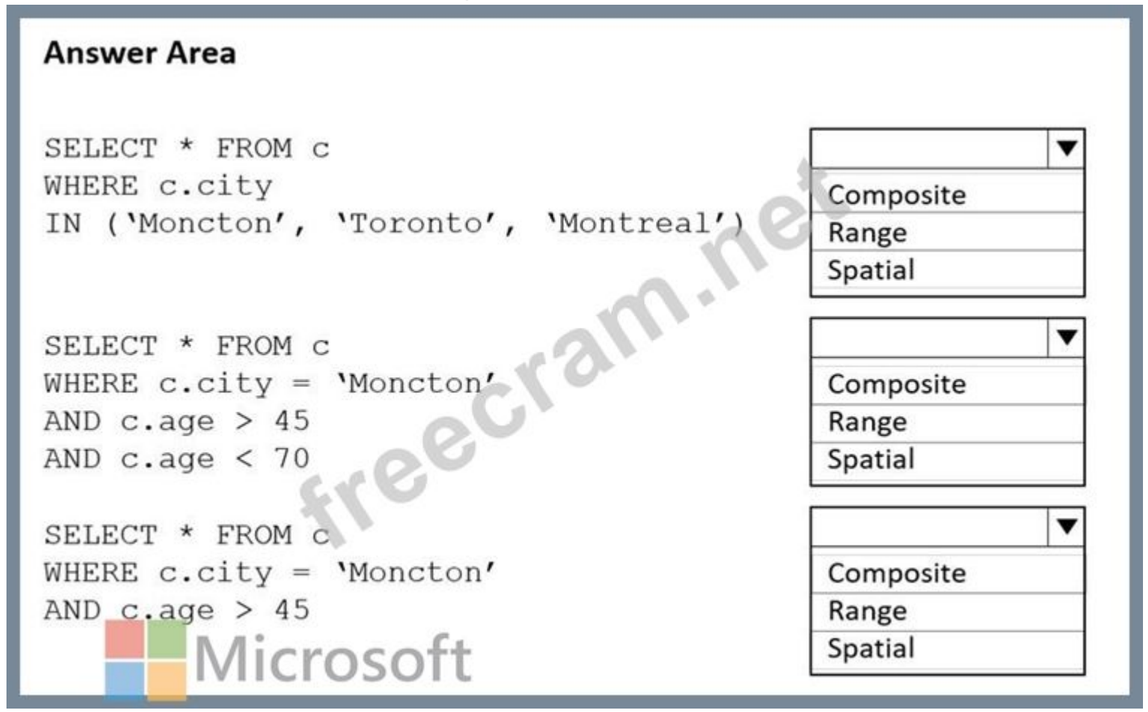
Statements	Yes	No
To return for all devices owned by a specific `emailaddress`, multiple queries must be performed	○	○
To return `deviceid`, `ownerid`, `timestamp`, and `value` for a specific `metricid`, a join must be performed	○	○
To return `deviceid`, `ownerid`, `emailaddress`, and `model`, a join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

Explanation:
Box 1: Yes
Need to join readings and devices.
Box 2: Yes
You need join to get the data from Array.
Box 3: No
Only devices is required. All required fields are in devices.

NEW QUESTION: 95
You have a container in an Azure Cosmos DB for NoSQL account.
You need to create an alert based on a custom Log Analytics query.
Which signal type should you use?
A. Resource Health
B. Activity Log
C. Log
D. Metrics
Answer: C (LEAVE A REPLY)

NEW QUESTION: 96
Hotspot Question
You have an Azure Cosmos DB for NoSQL account that frequently receives the same three queries.

You need to configure indexing to minimize RUs consumed by the queries.
Which type of index should you use for each query? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



freecram.net

Answer:

Answer Area

```
SELECT * FROM c
WHERE c.city
IN ('Moncton', 'Toronto', 'Montreal')
```

[Composite | Range | Spatial]

```
SELECT * FROM c
WHERE c.city = 'Moncton'
AND c.age > 45
AND c.age < 70
```

[Composite | Range | Spatial]

```
SELECT * FROM c
WHERE c.city = 'Moncton'
AND c.age > 45
```

[Composite | Range | Spatial]

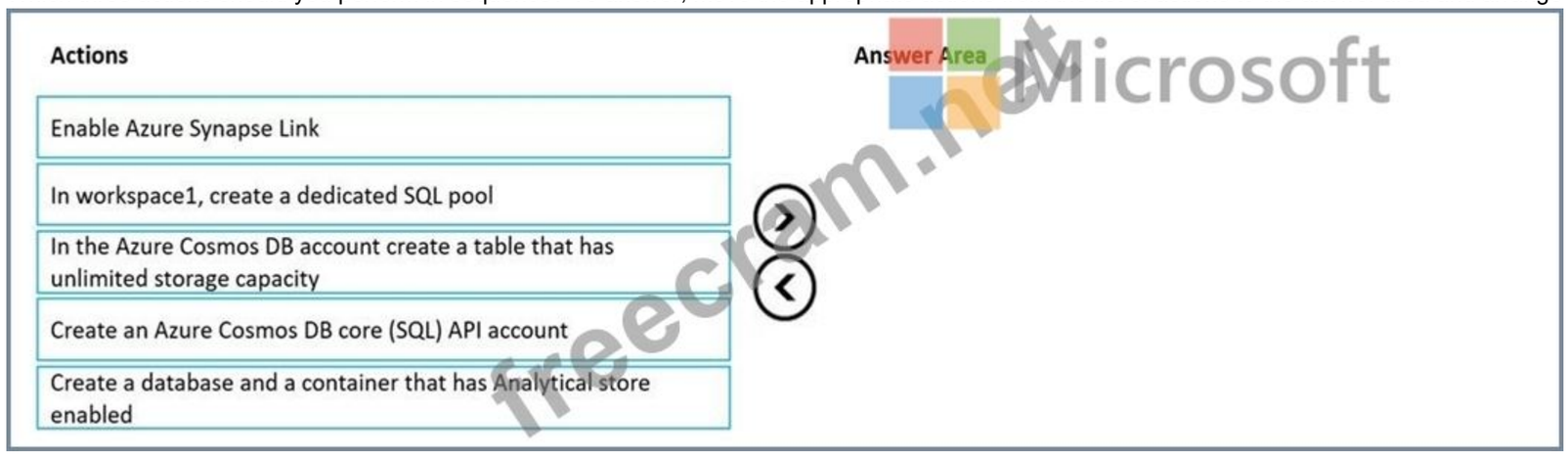
Microsoft
freecram.net

Explanation:
Box 1: Range
Range index is based on an ordered tree-like structure. The range index type is used for:
Equality queries:
SELECT * FROM container c WHERE c.property = 'value'
SELECT * FROM c WHERE c.property IN ("value1", "value2", "value3")
Box 2: Composite
Composite indexes increase the efficiency when you are performing operations on multiple fields.
The composite index type is used for:
Queries with a filter on two or more properties where at least one property is an equality filter SELECT * FROM container c WHERE c.property1 = 'value' AND c.property2 > 'value' Box 3: Composite Incorrect:
Spatial indices enable efficient queries on geospatial objects such as - points, lines, polygons, and multipolygon. These queries use ST_DISTANCE, ST_WITHIN, ST_INTERSECTS keywords.
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/index-overview

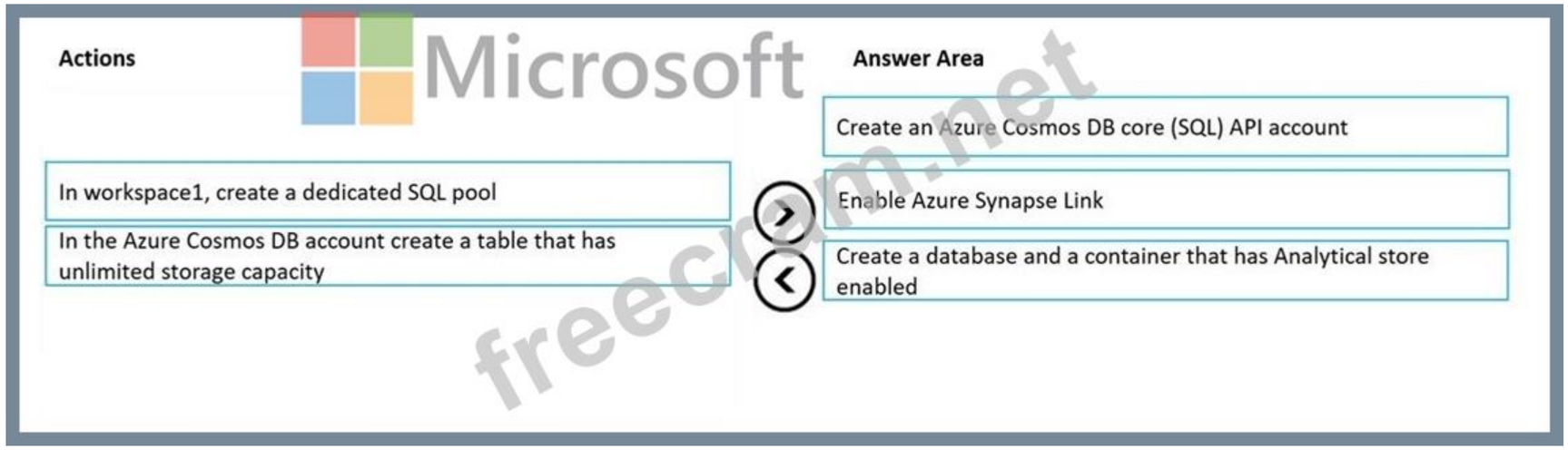
NEW QUESTION: 97
You have operational data in an Azure Cosmos DB for NoSQL database.
Database users report that the performance of the database degrades significantly when a business analytics team runs large Apache Spark-based queries against the database.
You need to reduce the impact that running the Spark-based queries has on the database users.
What should you implement?
A. Azure Synapse Link
B. a default consistency level of Consistent Prefix
C. a default consistency level of Strong
D. the Spark connector

Answer: (SHOW ANSWER)
Azure Synapse Link for Cosmos DB creates a tight integration between Azure Cosmos DB and Azure Synapse Analytics. It allows you to run large-scale analytics on operational data without affecting the performance of the operational workloads in Cosmos DB.
Synapse Link uses a columnar store to replicate data from Cosmos DB. This replication is asynchronous and does not impact the transactional store's performance, making it ideal for scenarios where operational workloads and analytical workloads need to coexist without impacting each other.

NEW QUESTION: 98
Drag and Drop Question
You have an Azure Synapse Analytics workspace named workspace1 that contains a serverless SQL pool.
You have an Azure Table Storage account that stores operational data.
You need to replace the Table storage account with Azure Cosmos DB for NoSQL. The solution must meet the following requirements:
- Support queries from the serverless SQL pool.
- Only pay for analytical compute when running queries.
- Ensure that analytical processes do NOT affect operational processes.
Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.



Answer:



Microsoft
freecram.net

Explanation:

Step 1: Create an Azure Cosmos DB core (SQL) API account

Step 2: Enable Azure Synapse Link

Synapse Link creates a tight seamless integration between Azure Cosmos DB and Azure Synapse Analytics.

Serverless SQL pool allows you to query and analyze data in your Azure Cosmos DB containers that are enabled with Azure Synapse Link. You can analyze data in near real-time without impacting the performance of your transactional workloads.

Step 3: Create a database and a container that has Analytical store enabled Create an analytical store enabled container Reference:

https://docs.microsoft.com/en-us/azure/cosmos-db/configure-synapse-link

NEW QUESTION: 99

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.

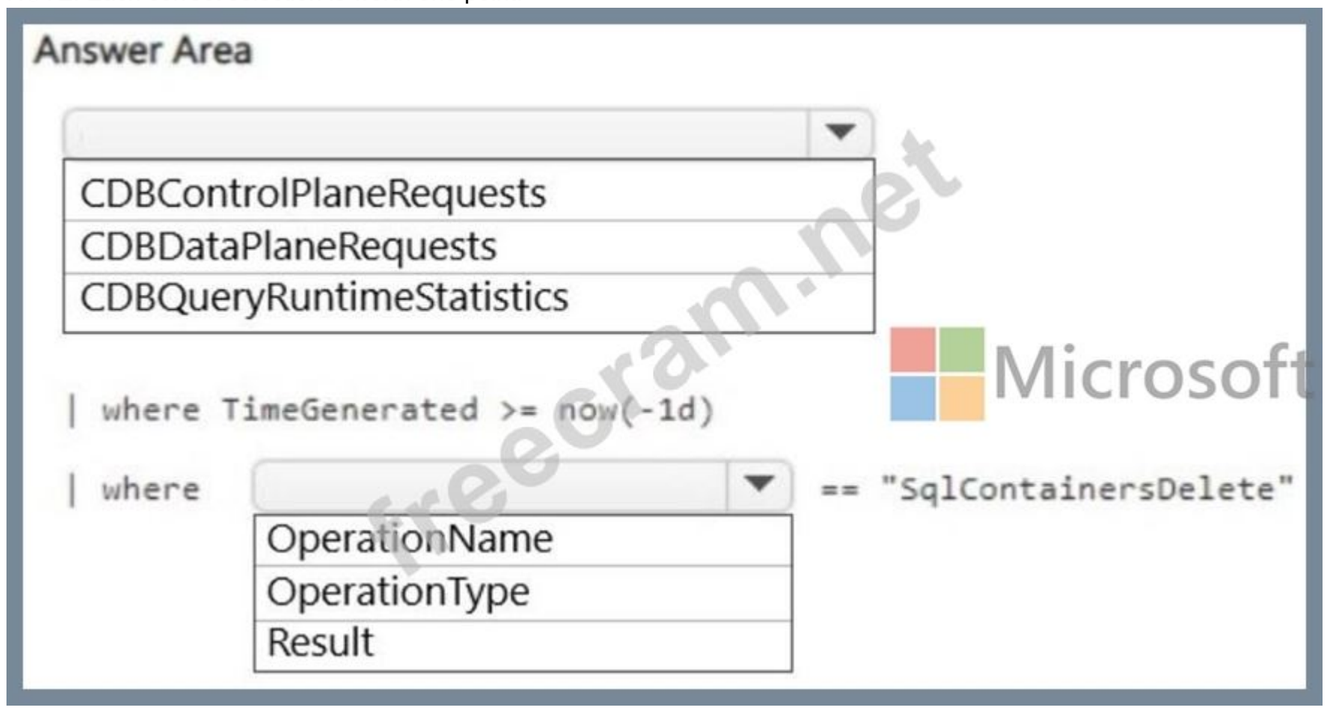
C. In a user container, create a document for each user. Embed the company into each user document. Use the user ID as the partition key.

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: (SHOW ANSWER)

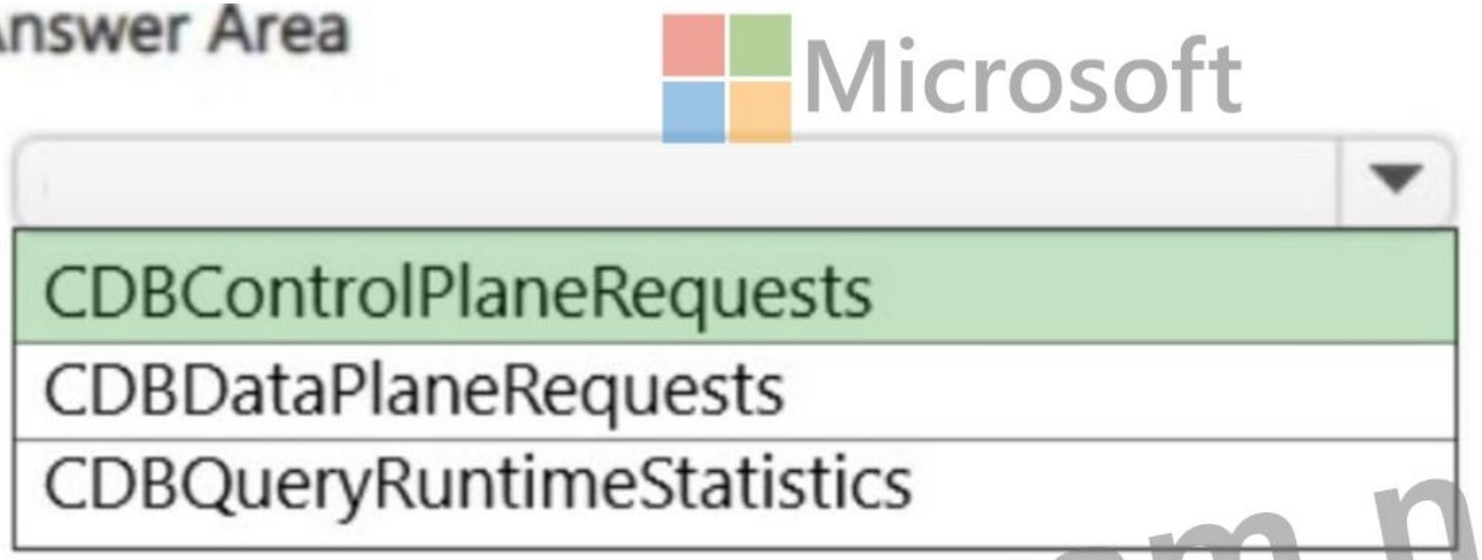
All employees within a company would nicely fit within a single document (document size 2 MB).
Note: An Azure Cosmos container is the unit of scalability both for provisioned throughput and storage. A container is horizontally partitioned and then replicated across multiple regions. The items that you 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

NEW QUESTION: 100
Hotspot Question
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account named account!
You configure account1 to send resource logs to Log Analytics.
You need to identify any containers in account1 that were deleted during the past 24 hours.
How should you complete the query? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:

Answer Area



freecram.net

NEW QUESTION: 101

You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account. The account hosts a container named Container1.

At some point during the past 4S hours, Container1 was deleted.

You need to restore the account to immediately before the deletion of Container1. The solution must minimize administrative effort.

What should you do first?

A. Review the event feed blade.

B. Identify the last restorable timestamp.

C. Create a new Azure Cosmos DB for NoSQL account.
D. Restore Container1 to a different Azure Cosmos DB for NoSQL account.
Answer: (SHOW ANSWER)
Azure Cosmos DB provides a point-in-time restore (PITR) feature, which allows you to restore data to a previous state. To restore a deleted container like Container1, the first step is to identify the last restorable timestamp.
This timestamp would be the point in time immediately before the deletion of the container.
Once you have identified the correct timestamp, you can proceed with the restore process.
Cosmos DB allows you to restore the container to either the same or a different account, but identifying the timestamp is the crucial first step.

NEW QUESTION: 102
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: (SHOW ANSWER)
The ItemRequestOptions class contains various session token and consistency level configuration properties on a per-request basis.

NEW QUESTION: 103
You have an on-premises computer named Computer1 that runs Windows 11.
On Computer1, you install the Azure Cosmos DB Emulator by using the default settings.
You need to connect to the API for NoSQL clients hosted by the emulator.
What should you use?
A. Computer1 and the credentials used when installing the emulator
B. localhost:443 and the built-in Administrator user account credentials
C. Computer1 and a randomly-generated key
D. localhost:8081 and a well-known key
Answer: (SHOW ANSWER)
Develop locally using the Azure Cosmos DB emulator
Create a new instance of CosmosClient using the emulator's credentials.
In C#
using CosmosClient client = new(
accountEndpoint: "https://localhost:8081/",
authKeyOrResourceToken:
"C2y6yDjf5/R+ob0N8A7Cgv30VRDJIWEHLM+4QDU5DE2nQ9nDuVTqobD4b8mGGyPMbIZnqy MsEcaGQy67XIw/Jw=="); Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/how-to-develop-emulator

NEW QUESTION: 104
You have an Azure subscription. The subscription contains an Azure Cosmos DB for NoSQL account named account1 that uses strong consistency and is deployed to the East US, West US, and Central US Azure regions.
The round-trip time (RTT) between the regions is shown in the following table.

Source region	Target region	RTT (ms)
East US	Central US	96
East US	West US	136
Central US	West US	54

freecram.net

Which write latency should you estimate for write requests against account1?

A. 150 ms

B. 232 ms

C. 282 ms

D. 572 ms

Answer: (SHOW ANSWER)

Calculate two times the maximum round-trip time (RTT) + 10 ms overhead.

In this scenario this would be 2 x 136 ms + 10 ms, which is 282 ms.

Note: For an Azure Cosmos DB account with strong consistency and multiple regions, write latency is approximately two times the round-trip time (RTT) between the two farthest regions.

This is because a write must be committed to all regions to be considered complete, requiring the data to be sent to the farthest region and then confirmed back to the originating region.

Calculation and factors Core calculation: The write latency is essentially 2 x RTT + constanst overhead. For example, if the RTT between the two farthest regions is 50 ms, the write latency would be roughly 2 x 50 ms + overhead=100 ms + overhead.

Overhead: In practice, you must also add a small constant overhead, which is approximately 10 ms at the 99th percentile for strong consistency, to account for the time it takes for the write to be acknowledged after the data reaches all replicas.

Reference:

https://learn.microsoft.com/en-us/azure/networking/azure-network-latency

NEW QUESTION: 105

Hotspot Question

You have an Azure subscription.

You need to create a container named Container1 in an Azure Cosmos DB for NoSQL account by using a Bicep file. The solution must ensure that Container1 supports partition key sizes of up to 1 KB.

How should you complete the file? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
...
partitionKey: {
  paths: [
    '/TenantId',
    '/UserId'
  ]
  kind:
}
  version:
}
...
```

kind: Hash | MultiHash | Range

version: 1 | 2 | 3

freecram.net

Microsoft

Answer:

Answer Area

```
...
partitionKey: {
  paths: [
    '/TenantId',
    '/UserId'
  ]
  kind:
  version:
}
...
```

kind: Hash | MultiHash | Range

version: 1 | 2 | 3

Microsoft
freecram.net

NEW QUESTION: 106
Hotspot Question
You have an Azure Cosmos DB for NoSQL account that hosts a container named Container1.
You are developing an Azure Function named Function1.
You need to ensure that Function1 will be triggered when an item is created or updated in Container1.
How should you complete the function? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
{
```

```
"type": "cosmosDBTrigger",
"name": "changes",
"direction": [ ▼ ],
```

- "change"
- "in"
- "out"
- "update"

```
"connectionStringSetting": "myconnectionstring",
"databaseName": "products",
"collectionName": "cars",
[ ▼ ] : "feed",
```

- "binding"
- "leaseCollectionName"
- "partitionKey"
- "sqlQuery"

```
"createLeaseCollectionIfNotExists": false
}
```

Microsoft
freecram.net

Answer:

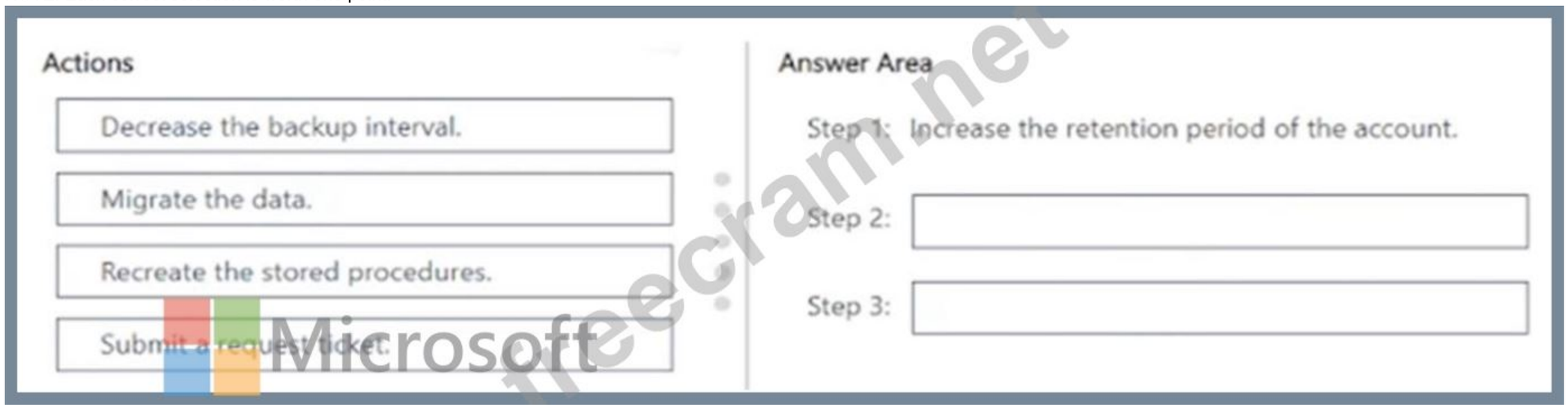
Answer Area



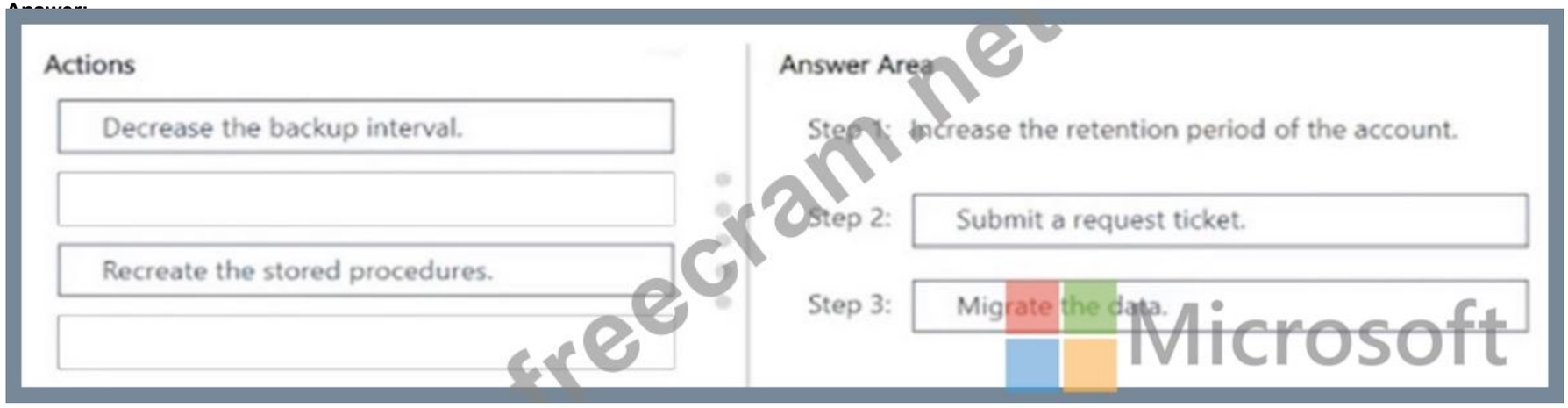
Valid DP-420 Dumps shared by EduDump.com for Helping Passing DP-420 Exam! EduDump.com now offer the **newest DP-420 exam dumps**, the EduDump.com DP-420 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com DP-420 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/DP-420/premium/ (**281** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 107
Drag and Drop Question
You have an Azure Cosmos DB for NoSQL account named account1 that has a periodic backup policy with a four-hour backup interval and an eight-hour retention period. The account1 account contains a single database

named db1 and containers that have stored procedures.
An administrator deletes db1 accidentally.
You need to ensure that the contents of db1, including all of its items, is available as soon as possible.
How should you perform the procedure? To answer, drag the missing actions to the correct order.
Each action may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.
NOTE: Each correct selection is worth one point.



Answer:



freecram.net
Microsoft

NEW QUESTION: 108
You have an Azure Cosmos DB account named account1.
You need to access account1 from an on-premises environment by using a Site-to-Site VPN.
What should you use?
A. a private endpoint
B. a dedicated gateway

C. Azure Synapse Link

Answer: (SHOW ANSWER)

A private endpoint allows you to access Azure services, including Azure Cosmos DB, privately from within an Azure virtual network. By configuring a private endpoint for the Azure Cosmos DB account, you can connect to the account using private IP addresses and ensure that the data transfer occurs securely over the Site-to-Site VPN between the on-premises network and the Azure virtual network.

With a private endpoint, the data traffic from the on-premises environment to Azure Cosmos DB remains within the private network and does not traverse the public internet, enhancing security and isolation.

NEW QUESTION: 109

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

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 (LEAVE A REPLY)

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 finction2 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

NEW QUESTION: 110

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

You have a web service that receives JSON-based messages.

You need to recommend a solution that will process and store the messages in a container in account1.

The solution must minimize latency.

What should you include in the recommendation?

Freecram.net

A. Azure Data Factory

B. Azure Synapse pipelines

C. an Azure SQL Managed Instance link

D. Azure Stream Analytics

Answer: (SHOW ANSWER)

To process and store JSON messages in Azure Cosmos DB with the lowest latency, you should use Azure Stream Analytics.

Real-time processing: It processes streaming data with sub-second latencies.

Native integration: It has built-in outputs for the Azure Cosmos DB Core (SQL) API.JSON support: It natively parses and processes JSON payloads without complex code.

Incorrect:

[Not A, Not B]
Azure Data Factory / Synapse Pipelines: These are built for batch processing and scheduling.
They introduce minutes of latency, making them too slow for real-time needs.
[Not C]
Azure SQL Managed Instance Link: This tool replicates data between SQL Server and Azure SQL Managed Instance. It is completely irrelevant for moving web service data into Cosmos DB.
Reference:
https://learn.microsoft.com/en-us/azure/stream-analytics/stream-analytics-documentdb-output

NEW QUESTION: 111
Hotspot Question
You have the Azure Cosmos DB for NoSQL containers shown in the following table.

Name	DefaultTimeToLive
container1	-1
container2	null
container3	60

You have the items shown in the following table.

Name	Container	TimeToLive
item1	container1	60
item2	container2	10
item3	container3	-1

When will each item expire? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

Item1:
- Never
- 10 seconds
- 60 seconds
- 10 minutes
- 60 minutes

Item2:
- Never
- 10 seconds
- 60 seconds
- 10 minutes
- 60 minutes

Item3:
- Never
- 10 seconds
- 60 seconds
- 10 minutes
- 60 minutes

freecram.net

Answer:

Answer Area



Microsoft
freecram.net

Explanation:
https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/time-to-live

NEW QUESTION: 112
You have an Azure Cosmos DB for NoSQL database that hosts a container named Container1.
For Container1, the DefaultTimeToLive property is -1.
You insert an item into Container1. The ttl property of the inserted item is 1000.
When will the item be deleted automatically?
A. one second after the item is last accessed
B. 1,000 seconds after the item is last accessed
C. 1,000 seconds after the item is last modified
D. never
Answer: (SHOW ANSWER)
The item will be deleted after 1,000 seconds (approximately 16.7 minutes) from the time it was last modified.
In Azure Cosmos DB, the Time to Live (TTL) logic follows a specific hierarchy of precedence:
TTL Logic Breakdown

Container Level (DefaultTimeToLive = -1): This setting enables TTL for the container but instructs Cosmos DB to never expire items by default.
Item Level (ttl = 1000): When an individual item has a ttl property, it overrides the container-level setting.
The Result: Because the item-level value is a positive integer, the system honors that specific duration for that specific item.
Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/how-to-time-to-live

NEW QUESTION: 113
You plan to create an Azure Cosmos DB account that will use the NoSQL API.
You need to create a grouping strategy for items that will be stored in the account. The solution must ensure that write and read operations on the items can be performed within the same transaction.
What should you use to group the items?
A. logical partitions
B. physical partitions
C. databases
D. containers
Answer: (SHOW ANSWER)
Azure Cosmos DB supports full ACID compliant transactions with snapshot isolation for operations within the same logical partition key.
https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/how-to-manage-consistency?tabs=portal%2Cdotnetv2%2Capi-async#override-the-default-consistency-level

NEW QUESTION: 114
You have an Azure Cosmos DB for NoSQL account.
You configure the diagnostic settings to send all log information to a Log Analytics workspace.
You need to identify when the provisioned request units per second (RU/s) for resources within the account were modified.
You write the following query.
AzureDiagnostics
| where Category == "ControlPlaneRequests"
What should you include in the query?
A. | where OperationName startswith "SqlContainersThroughputUpdate"
B. | where OperationName startswith "SqlContainersCreate"
C. | where OperationName startswith "RegionAddComplete"
D. | where OperationName startswith "MongoCollectionsThroughputUpdate"
Answer: (SHOW ANSWER)

NEW QUESTION: 115
Hotspot Question
You have an Azure Cosmos DB for NoSQL account that hosts a container named Customers.
You need to add customers to Customers by using the Azure Cosmos DB Spark Connector.
How should you complete the code? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
newCustomers = spark.createDateFrame((\
     ("ABC001", "ABC Corporation", 0, True),\
     ("BEL001", "BEL Corporation", 20000, False)))\
          .toDF("id", "companyName", "creditLimit", "isOnCreditHold"
newCustomers.write\
     .format
          ("cosmos.oltp")\
          ("CSV")\
          ("JSON")\
          ("PARQUET")\
     .options(**cfg)\
     .mode
          ("APPEND")\
          ("cosmos.oltp")\
          ("INSERT")\
          ("JSON")\
          ("PARQUET")\
     .save()
```

Microsoft freecram.net

Answer:

Answer Area

```
newCustomers = spark.createDateFrame((\
     ("ABC001", "ABC Corporation", 0, True),\
     ("BEL001", "BEL Corporation", 20000, False)))\
          .toDF("id", "companyName", "creditLimit", "isOnCreditHold"
newCustomers.write\
     .format
          ("cosmos.oltp")\
          ("CSV")\
          ("JSON")\
          ("PARQUET")\
     .options(**cfg)\
     .mode
          ("APPEND")\
          ("cosmos.oltp")\
          ("INSERT")\
          ("JSON")\
          ("PARQUET")\
     .save()
```

Microsoft freecram.net

NEW QUESTION: 116

You are designing an Azure Cosmos DB for NoSQL solution to store data from IoT devices.

Writes from the devices will occur every second. Data will be retained indefinitely.
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"
}
```

freecram.net



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 deviceId as the partition key.
B. Create a new synthetic key that contains deviceId and timestamp.
C. Use sensor1value as the partition key.
D. Create a new synthetic key that contains deviceId and sensor1Value.

Answer: (SHOW ANSWER)

Correct:
* Create a new synthetic key that contains deviceId and a random number.
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.
* Create a new synthetic key that contains deviceId and timestamp.
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:
* Create a new synthetic key that contains deviceId and deviceManufacturer.
All the devices could have the same manufacturer.
* Create a new synthetic key that contains deviceId and sensor1Value.
Senser1Value has only two values.
* Use deviceId as the partition key.
* Use deviceManufacturer as the partition key.
All the devices could have the same manufacturer.
* Use sensor1Value as the partition key
* Use timestamp as the partition key.
You will also not like to partition the data on "DateTime", because this will create a hot partition.
Imagine you have partitioned the data on time, then for a given minute, all the calls will hit one partition. If you need to retrieve the data for a customer, then it will be a fan-out query because data may be distributed on all the partitions.
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

NEW QUESTION: 117
You have an Azure Cosmos DB for NoSQL database that hosts a container named Container1.
For Container1, the DefaultTimeToLive property is -1.
You insert an item into Container1. The ttl property of the inserted item is 1000.
When will the item be deleted automatically?
A. 1,000 seconds after the item is last modified
B. 1,000 seconds after the item is last accessed
C. one second after the item is last accessed
D. never
Answer: (SHOW ANSWER)

NEW QUESTION: 118
You have a container in an Azure Cosmos DB for NoSQL account.
Data update volumes are unpredictable.
You need to process the change feed of the container by using a web app that has multiple instances. The change feed will be processed by using the change feed processor from the Azure Cosmos DB SDK. The multiple instances must share the workload.
Which three actions should you perform? Each correct answer presents part of the solution.
NOTE: Each correct selection is worth one point.

A. Configure the same processor name for all the instances
B. Configure a different processor name for each instance
C. Configure a different instance name for each instance
D. Configure a different lease container configuration for each instance
E. Configure the same instance name for all the instances
F. Configure the same lease container configuration for all the instances
Answer: (SHOW ANSWER)
A: Implementing the change feed processor.
The point of entry is always the monitored container, from a Container instance you call GetChangeFeedProcessorBuilder.
C: You define the compute instance name or unique identifier with WithInstanceName, this should be unique and different in each compute instance you are deploying.
F: The lease container: The lease container acts as a state storage and coordinates processing the change feed across multiple workers. The lease container can be stored in the same account as the monitored container or in a separate account.
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/sql/change-feed-processor

NEW QUESTION: 119
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account named account1.
Backups for account1 have the following configurations:
- Interval: 2 hours
- Retention period: 4 days
You need to estimate the charges associated with the retention of the backups.
How many copies of the backups will incur additional charges?
A. 48
B. 8
C. 12
D. 46
Answer: (SHOW ANSWER)

NEW QUESTION: 120
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 a container named container1 in an Azure Cosmos DB for NoSQL account.
You need to make the contents of container1 available as reference data for an Azure Stream Analytics job.
Solution: You create an Azure Synapse pipeline that uses Azure Cosmos DB for NoSQL change feed as a trigger and an Azure event hub as the output.
Does this meet the goal?
Freecram.net
A. Yes
B. No
Answer: (SHOW ANSWER)

NEW QUESTION: 121

Hotspot Question

You have the following Azure Resource Manager (ARM) template.

```
{
    "type": "Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers",
    "apiVersion": "2021-01-15",
    "name": "acct01/mydb/mycontainer",
    "properties": {
      "resource": {
        "id": "mycontainer",
        "partitionKey": {
          "paths": [
            "/companyid"
          ],
          "kind": "Hash"
        },
        "indexingMode": "consistent",
        "includedPaths": [
          {
            "path": "/*"
          },
          {
            "path": "/headquarters/country/?"
          }
        ],
        "excludedPaths": [
          {
            "path": "/headquarters/*"
          }
        ],
        "compositeIndexes": [
          [
            {
              "path": "/name",
              "order": "ascending"
            },
            {
              "path": "/startDate",
              "order": "descending"
```

Microsoft
freecram.net

```
                "order": "descending"
            }
        ]
    ]
  }
 }
}
```

You plan to deploy the template in incremental mode.

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:



Explanation:

Box 1: No

An account is not a resource in this context.

Note: In incremental mode, Resource Manager leaves unchanged resources that exist in the resource group but aren't specified in the template. Resources in the template are added to the resource group.
The Microsoft.DocumentDB databaseAccounts/sqlDatabases/containers resource type can be deployed to: Resource groups.
Box 2: Yes
A container is a resource.
Box 3: No
Note that you can't change the partition key of a container. If you do need to change it, you need to migrate the container data to a new container with the correct key.
Reference:
https://docs.microsoft.com/en-us/azure/templates/microsoft.documentdb/databaseaccounts/sqldatabases/containers
https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deployment-modes#incremental-mode

Valid DP-420 Dumps shared by EduDump.com for Helping Passing DP-420 Exam! EduDump.com now offer the **newest DP-420 exam dumps**, the EduDump.com DP-420 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com DP-420 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/DP-420/premium/ (**281** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 122
Hotspot Question
You have an Azure Cosmos DB for NoSQL 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
```

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	○	○

Microsoft
freecram.net

Answer:



Microsoft
freecram.net

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: 123

You develop an application that uses Azure Cosmos DB Core (SQL) API.

You create an Azure pipeline to build and deploy the application.

You need to change the pipeline to run integration tests that you wrote for the application. The solution must execute entirely in the pipeline.

What should you add to the pipeline?

A. a NuGet service connection that uses an Azure Cosmos DB API key

B. an Azure Cosmos DB Emulator task

C. a deployment group named Cosmos DB testing

D. a secret variable that has a connection string to an Azure Cosmos DB database

Answer: (SHOW ANSWER)

NEW QUESTION: 124

Case Study 1 - Litware, inc

Overview

Litware, Inc. is a United States-based grocery retailer. Litware has a main office and a primary datacenter in Seattle. The company has 50 retail stores across the United States and an emerging online presence. Each store connects directly to the internet. Existing environment.

Cloud and Data Service Environments. Litware has an Azure subscription that contains the resources shown in the following table.

Name	Type	Description
account1	Azure Cosmos DB for NoSQL account	The account has a single read-write region and contains a database named productdb. The productdb database contains two containers named con-product and con-productVendor. The account uses the session default consistency level.
iothub1	Azure IoT hub	The IoT hub collects per-second temperature and humidity telemetry from 5,000 IoT devices in the retail stores. A single telemetry reading generates 1 KB of data.
streamanalytics1	Azure Stream Analytics job	The job processes telemetry data collected from iothub1.

Each container in productdb is configured for manual throughput. The con-product container stores the company's product catalog data. Each document in con-product includes a con- productvendor value. Most queries targeting the data in con-product are in the following format.

SELECT * FROM con-product p WHERE p.con-productVendor - 'name'

Most queries targeting the data in the con-productVendor container are in the following format SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Existing environment.

Current Problems.

Litware identifies the following issues:

- Updates to product categories in the con-productVendor container do not propagate automatically to documents in the con-product container.
- Application updates in con-product frequently cause HTTP status code 429 "Too many requests".
- You discover that the 429 status code relates to excessive request unit (RU) consumption during the updates.

Requirements.

Planned Changes

Litware plans to implement a new Azure Cosmos DB for NoSQL account named account2 that will contain a database named iotdb. The iotdb database will contain two containers named con- iot1 and con-iot2.

Litware plans to make the following changes:

- Store the telemetry data in account2.
- Configure account1 to support multiple read-write regions.
- Implement referential integrity for the con-product container.
- Use Azure Functions to send notifications about product updates to different recipients.
- Develop an app named App1 that will run from all locations and query the data in account1.
- Develop an app named App2 that will run from the retail stores and query the data in account2.
- App2 must be limited to a single DNS endpoint when accessing account2.

Requirements.

Business Requirements

Litware identifies the following business requirements:

- 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.
- Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.
- Minimize the number of firewall changes in the retail stores.

Requirements.

Product Catalog Requirements

Litware identifies the following requirements for the product catalog:

- Implement a custom conflict resolution policy for the product catalog data.
- Minimize the frequency of errors during updates of the con-product container.
- Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.

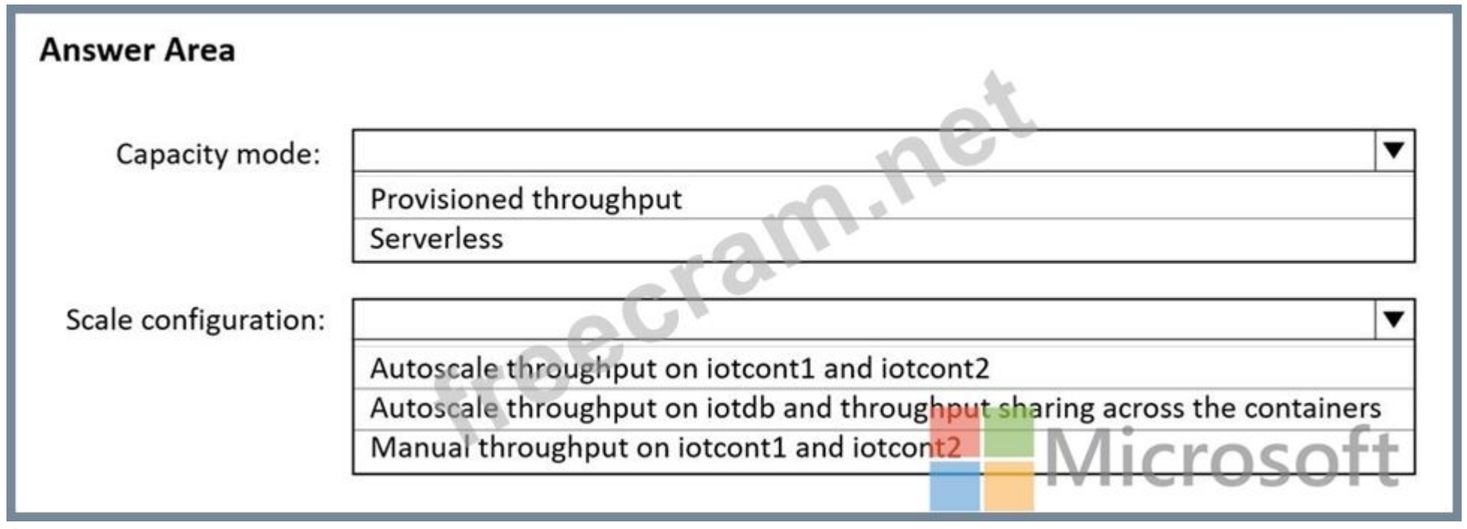
- Trigger the execution of two Azure functions following every update to any document in the con- product container.

Hotspot Question

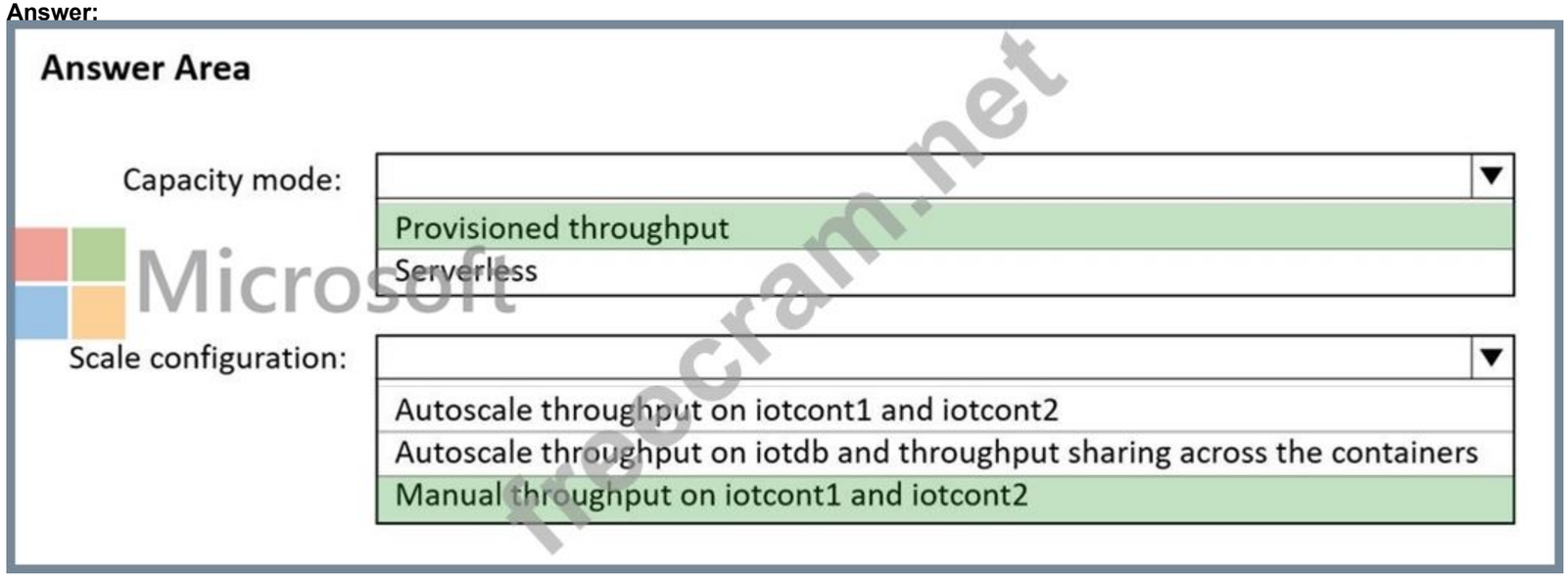
You need to select the capacity mode and scale configuration for account2 to support the planned changes and meet the business requirements.

What should you select? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.



Answer:



Explanation:

Box 1: Provisioned throughput

Provisioned throughput is best suited for workloads with sustained traffic requiring predictable performance.

Billing model: Billing is done on a per-hour basis for the RU/s provisioned, regardless of how many RUs were consumed.

Scenario: Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.

Note: Azure Cosmos DB is available in two different capacity modes: provisioned throughput and serverless. You can perform the exact same database operations in both modes, but the way you get billed for these operations is radically different.
Box 2: Manual throughput on itocont1 and iotcont2
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/throughput-serverless
https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-provision-autoscale-throughput

NEW QUESTION: 125
Hotspot Question
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account.
You create a container and assign the following indexing policy.

```
{
    "automatic":true,
    "indexingMode":"consistent",
    "includedPaths":[
        {
            "path":"/*"
        }
    ],
    "excludedPaths":[],
    "compositeIndexes":[
        [
            {
                "path":"/name",
                "order":"ascending"
            },
            {
                "path":"/age",
                "order":"descending"
            }
        ]
    ]
}
```

Microsoft
freecram.net

You run the queries shown in the following table.

Name	Query
Query 1	SELECT name, age, city FROM c ORDER BY c.name, c.age DESC
Query 2	SELECT name, age, city FROM c ORDER BY c.name DESC, c.age
Query 3	SELECT name, age, city FROM c WHERE c.age = 18 AND c.city = "Boston" ORDER BY c.name DESC, c.age

freecram.net

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
Query 1 can use the defined indexes.	○	○
Query 2 can use the defined indexes.	○	○
Query 3 can use the defined indexes.	○	○

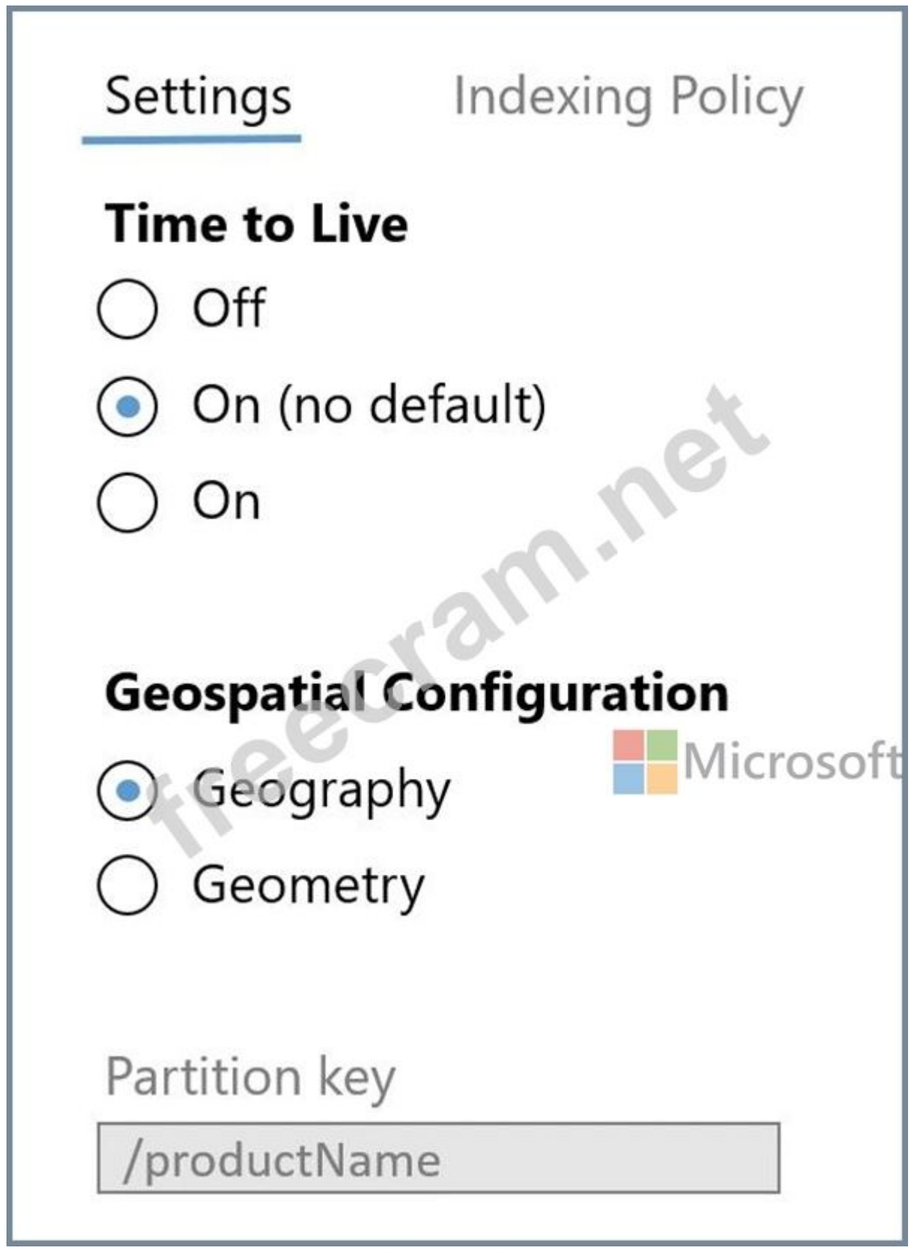
Answer:

Answer Area

Statements	Yes	No
Query 1 can use the defined indexes.	○	○
Query 2 can use the defined indexes.	○	○
Query 3 can use the defined indexes.	○	○

NEW QUESTION: 126
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 a container named container1 in an Azure Cosmos DB for NoSQL account.
You need to make the contents of container1 available as reference data for an Azure Stream Analytics job.
Solution: You create an Azure function to copy data to another Azure Cosmos DB for NoSQL container.
Does this meet the goal?
A. Yes
B. No
Answer: (SHOW ANSWER)

NEW QUESTION: 127
The settings for a container in an Azure Cosmos DB for NoSQL account are configured as shown in the following exhibit.



Which statement describes the configuration of the container?

A. All items will be deleted after one year.

B. Items stored in the collection will be retained always, regardless of the items time to live value.

C. Items stored in the collection will expire only if the item has a time to live value.

D. All items will be deleted after one hour.

Answer: (SHOW ANSWER)

When DefaultTimeToLive is -1 then your Time to Live setting is On (No default) Time to Live on a container, if present and the value is set to "-1", it is equal to infinity, and items don't expire by default.
Time to Live on an item:
This Property is applicable only if DefaultTimeToLive is present and it is not set to null for the parent container.
If present, it overrides the DefaultTimeToLive value of the parent container.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/sql/time-to-live

NEW QUESTION: 128
You are designing an Azure Cosmos DB for NoSQL data model that will contain authors, books the authors have written, and reviews of the books.
You need to maintain referential integrity between entities by using server-side transactional support.
What is the largest scope of the transactions you can use?
A. items in the same container
B. a single item
C. items in the same database
D. items in the same logical partition
Answer: A (LEAVE A REPLY)
Transactional batch operations in Azure Cosmos DB
Transactional batch describes a group of point operations that need to either succeed or fail together with the same partition key in a container. Operations are defined, added to the batch, and the batch is executed. If all operations succeed in the order they're described within the transactional batch operation, the transaction will be committed. However, if any operation fails, the entire transaction is rolled back.
Does Azure Cosmos DB for NoSQL support ACID (atomicity, consistency, isolation, durability) transactions?
Yes, the API for NoSQL supports cross-document transactions expressed using either batches in the SDKs or as JavaScript-stored procedures and triggers. Transactions are scoped to a single partition within each container and executed with ACID semantics as "all or nothing," isolated from other concurrently executing code and user requests. If exceptions occur, the entire transaction is rolled back.
Reference:
https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/transactional-batch
https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/faq

NEW QUESTION: 129
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 IntegratedCacheiteItemHitRate 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: (SHOW ANSWER)
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

NEW QUESTION: 130

You are creating an Azure Cosmos DB for NoSQL account.
You need to choose whether to configure capacity by using either provisioned throughput or serverless mode.
Which two workload characteristics cause you to choose serverless mode? Each correct answer presents part of the solution.
NOTE: Each correct selection is worth one point.

Freecram.net

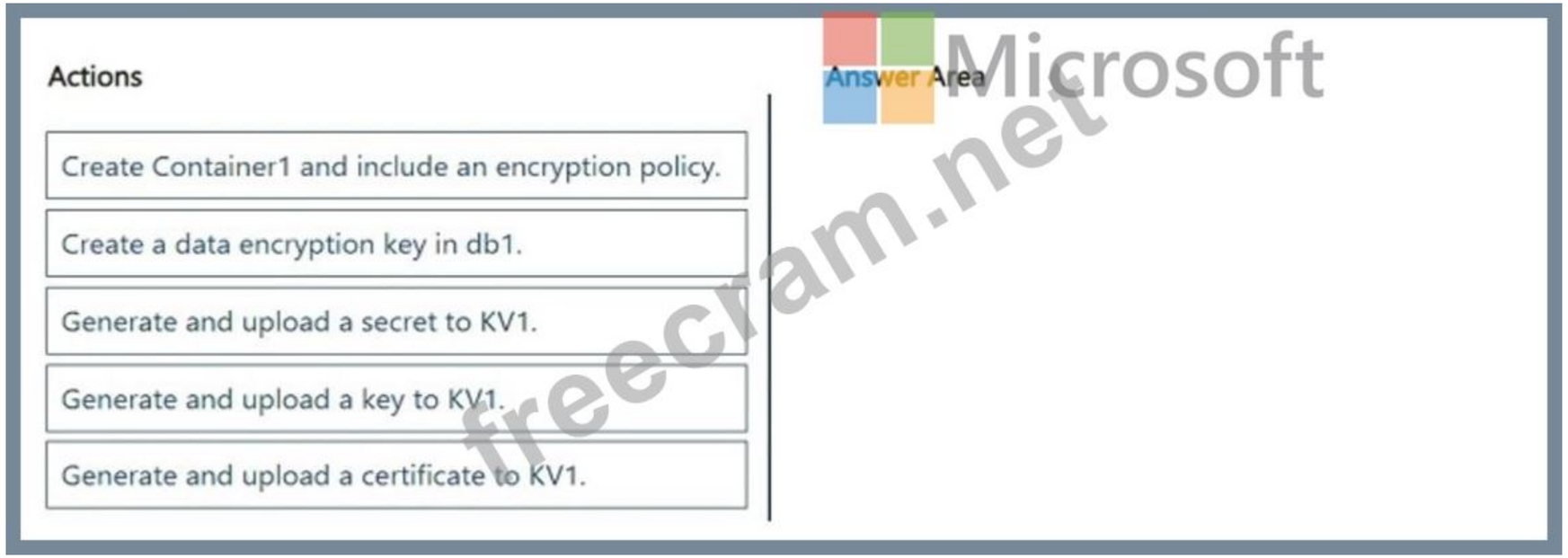
A. intermittent traffic
B. high average-to-peak traffic ratio
C. low average-to-peak traffic ratio
D. geo-distribution requirements
E. 25 TB of data in a single container
Answer: (SHOW ANSWER)

NEW QUESTION: 131
Drag and Drop Question
You have an Azure subscription that contains the resources shown in the following table.

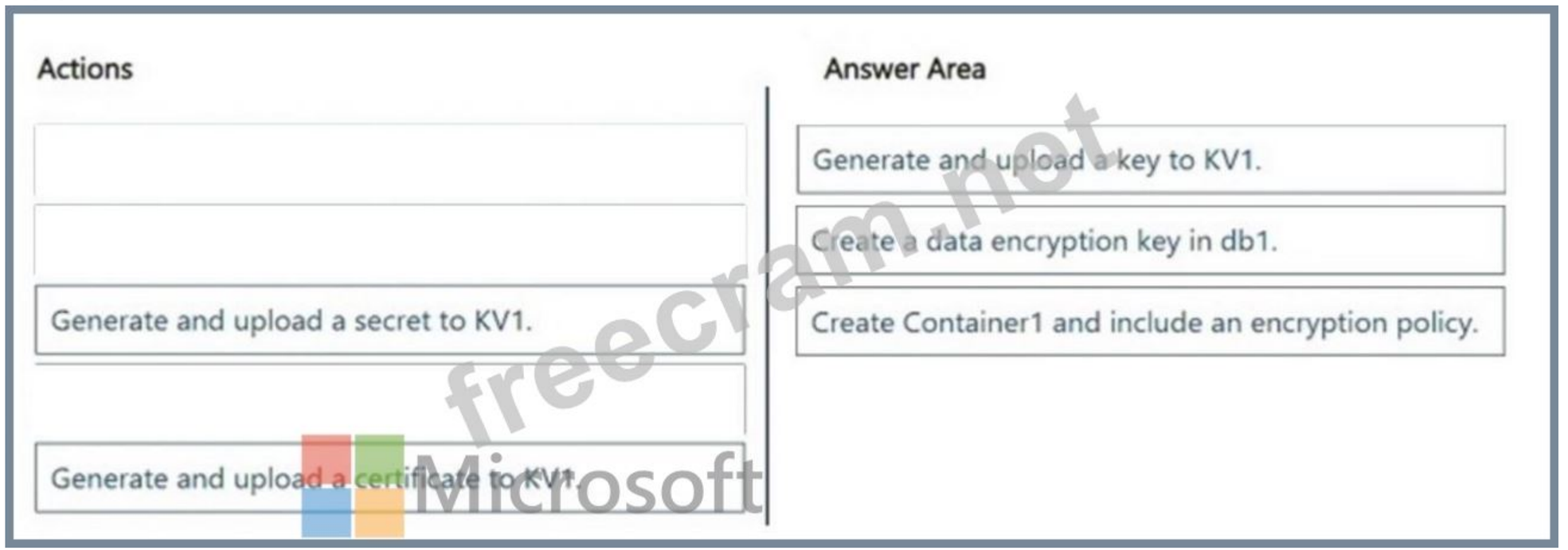
Name	Type	Description
account1	Azure Cosmos DB for NoSQL account	*None*
db1	Azure Cosmos DB for NoSQL database	Hosted in account1
KV1	Azure Key Vault	Has purge protection enabled

Freecram.net Microsoft

You need to create a container in db1 that has the following attributes:
- Name: Container1
- Encryption: Client-side encryption with Always Encrypted
Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.



Answer:



NEW QUESTION: 132

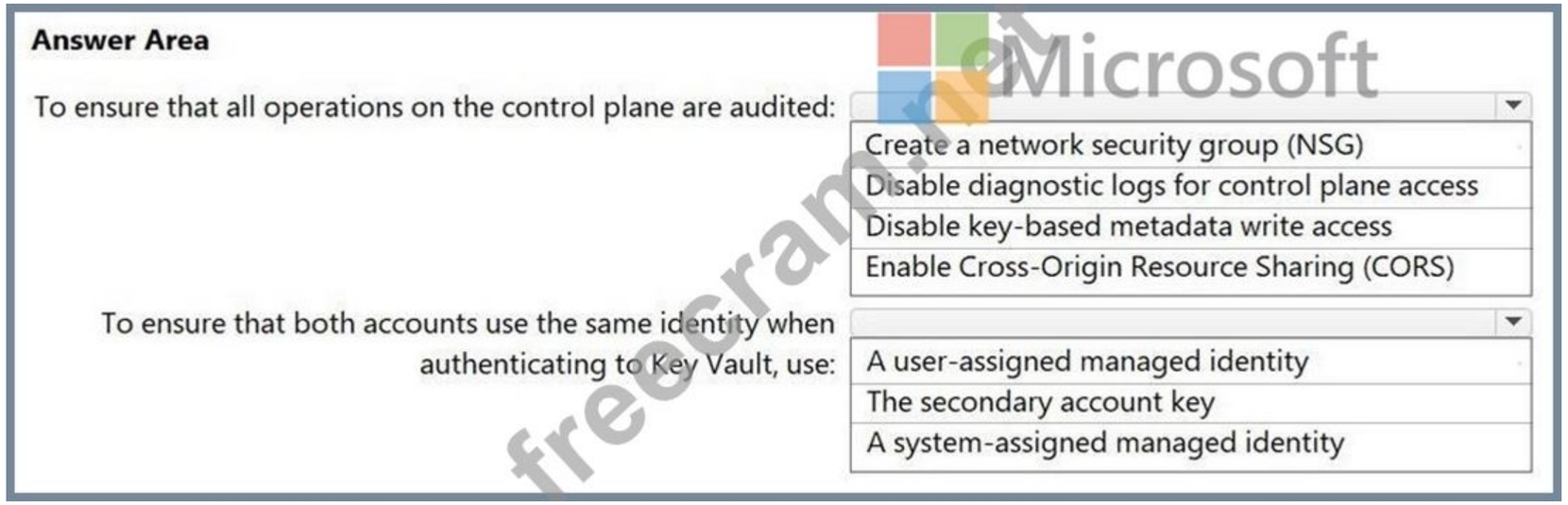
Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

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 for NoSQL 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: (SHOW ANSWER)
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: 133
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL database named DB1.
You develop a software as a service (SaaS) app named App1 that uses DB1. App1 will be used by multiple vendors to manage their customers.
You need to create a container in DB1 that will store customer and order details for each vendor.
The solution must meet the following requirements:
- Each vendor must have a unique tenant ID.
- The performance of the container must be optimized for reads.
- Customer IDs can be duplicated across the container but must be unique for each vendor.
- The container must be optimized for concurrent queries from multiple customers. Each query will return data for a single vendor.
How should you configure the container?
A. Add a partition key to the tenant ID and a unique key policy to the customer ID.
B. Add hierarchical partition key on (tenant identifier, customer identifier).
C. Add a partition key to the customer ID and a unique key policy to the tenant ID.
D. Add hierarchical partition key on (customer identifier, tenant identifier).
Answer: (SHOW ANSWER)
For this multi-tenant SaaS scenario, you should use a hierarchical partition key and a unique key policy.
Partitioning Strategy: Hierarchical Partition Key
A hierarchical partition key (subpartitioning) is the optimal choice for your requirements.
Structure: Define the hierarchy as /TenantId followed by /CustomerId.
Why it meets your needs:
Single Vendor Scope: Since every query returns data for a single vendor, including TenantId as the first level of the hierarchy ensures all related data is logically grouped and queries are efficiently routed to the correct partitions.
Scale Beyond 20 GB: Standard partition keys limit a single logical partition (e.g., one vendor) to 20 GB. Hierarchical partitioning allows a single TenantId to scale across multiple physical partitions, supporting larger vendors without hitting storage limits.
Read Optimization: It optimizes concurrent queries from multiple customers by distributing the workload of a single large vendor across different subpartitions.
Reference:

https://devblogs.microsoft.com/cosmosdb/scaling-multi-tenant-go-applications-choosing-the-right-database-partitioning-approach/

NEW QUESTION: 134
Hotspot Question
You plan to use two Azure Cosmos DB for NoSQL accounts. Both accounts will use Transparent Data Encryption (TDE).
You plan to store customer-managed keys in Azure Key Vault.
You need to configure security to meet the following requirements:
- Ensure that all operations on the control plane are audited.
- Ensure that both accounts use the same identity when authenticating
to Key Vault.
What should you do for each requirement? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:
Answer Area

To ensure that all operations on the control plane are audited:
- Create a network security group (NSG)
- Disable diagnostic logs for control plane access
- **Disable key-based metadata write access**
- Enable Cross-Origin Resource Sharing (CORS)

To ensure that both accounts use the same identity when authenticating to Key Vault, use:
- **A user-assigned managed identity**
- The secondary account key
- A system-assigned managed identity

NEW QUESTION: 135
Hotspot Question

You have a container that stores data about families.
The following is a sample document.

```
{
    "lastName":"Cartwright",
    "parents":[
      {
        "firstname":"Elvira",
        "role":"mother",
        "age":64
      },
      {
        "firstname":"Randolph",
        "role":"father",
        "age":67
      }
    ],
    "children":[
      {
        "grade":5,
        "pets":[

        ],
        "firstname":"Dana",
        "age":15,
        "gender":"female"
      },
      {
        "grade":7,
        "pets":[
          {
            "name":"Bob",
            "type":"guinea pig"
          },
```

freecram.net
Microsoft

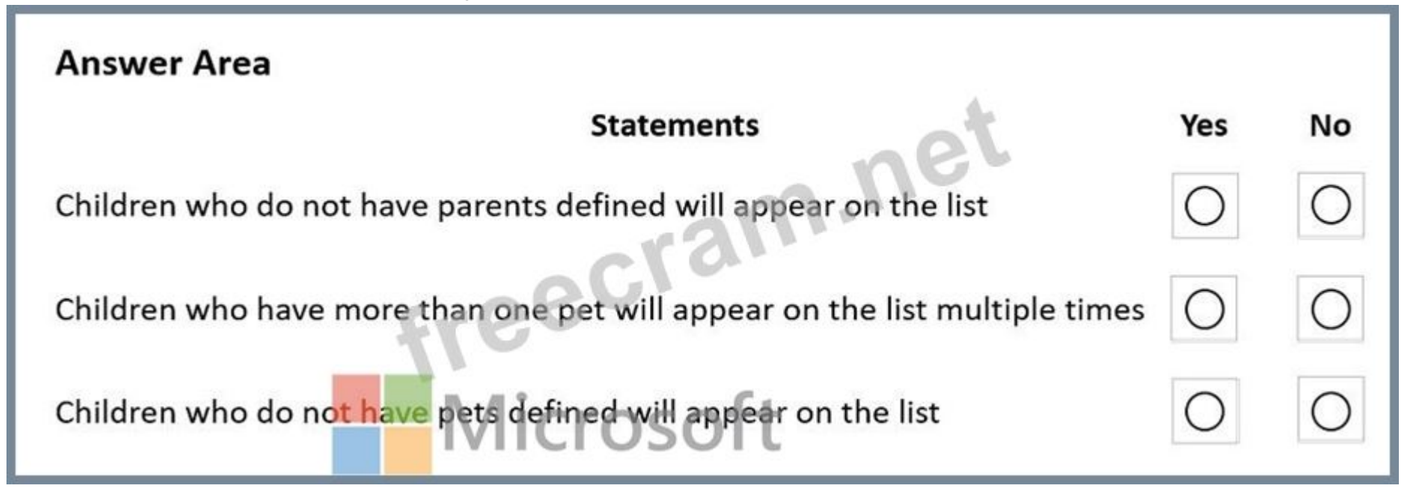
You run the following query against the container.

```
SELECT
    ch.name ?? ch.firstName AS childName,
    f.parents
    ARRAY_LENGTH(ch.pets) ?? 0 AS numberOfPets,
    ch.pets
FROM c AS f
JOIN ch IN f.children
```

Microsoft
freecram.net

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

NOTE: Each correct selection is worth one point.



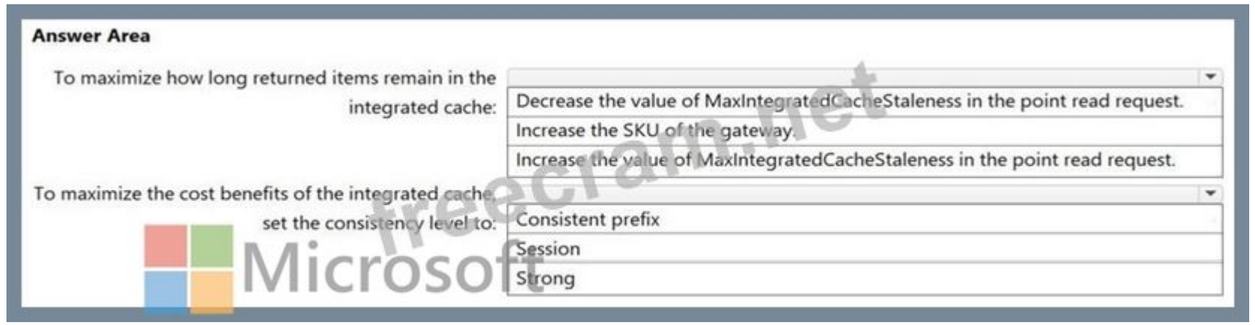
Answer:



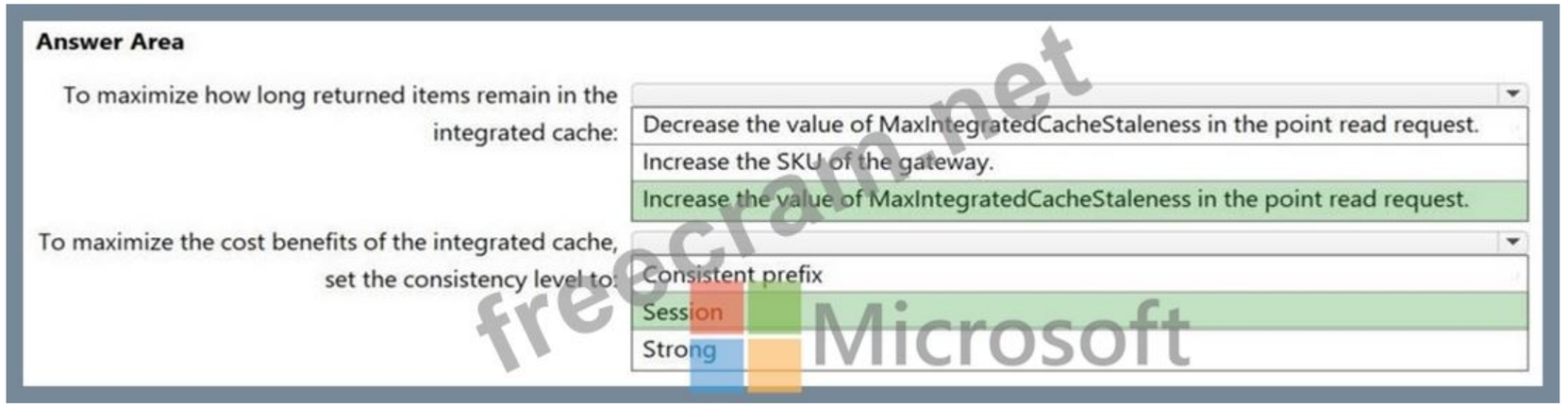
Explanation:
Box 1: Yes
The query return a value because "c" is represent the entire document and not only the parent's array.
Box 2: No
The cross join is not on the pets.
Box 3: Yes
The cross join is not on the pets.
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/sql/sql-query-join

NEW QUESTION: 136
Hotspot Question
You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account named account1.

For account1, you plan to implement an integrated gateway that will use the D4s SKU.
You need to optimize integrated gateway performance for point reads from account1. The solution must meet the following requirements:
- For new point reads, maximize how long returned items remain in the integrated cache.
- For new and repeating point reads, maximize the cost benefits of the integrated cache.
What should you do for each requirement? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:

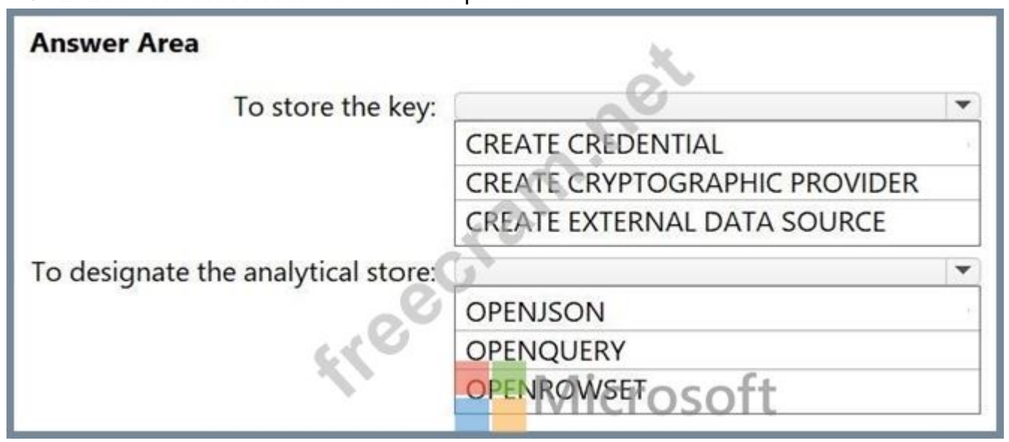


Valid DP-420 Dumps shared by EduDump.com for Helping Passing DP-420 Exam! EduDump.com now offer the **newest DP-420 exam dumps**, the EduDump.com DP-420 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com DP-420 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/DP-420/premium/ (**281** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

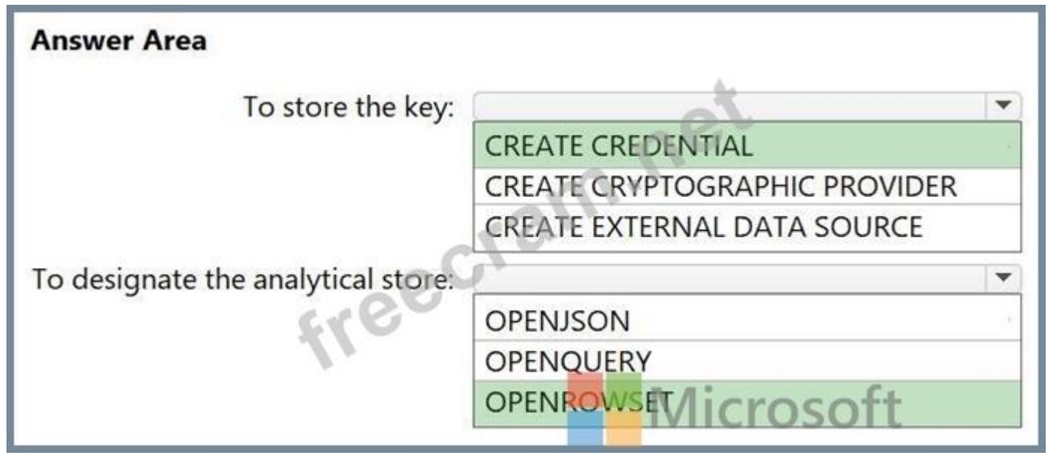
NEW QUESTION: 137
Hotspot Question
You have an Azure subscription that contains the resources shown in the following table.

Name	Type	Description
account1	Azure Cosmos DB for NoSQL account	*None*
DB1	Database	Hosted in account1
Container1	Container	Hosted in DB1 and has analytical store enabled
Synapse1	Azure Synapse Analytics instance	*None*

You plan to use multiple SELECT statements to query Container1 by using the built-in SQL pool of Synapse1.
You need to meet the following requirements:
- Store the key for account1 in Synapse1 for use in the SELECT
- Designate the analytical store in the FROM clause of the SELECT
What should you run? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:



NEW QUESTION: 138
You have an Azure subscription that hosts an Azure Cosmos DB for NoSQL container named Container1 and an Azure Data Factory pipeline named Pipeline1.
You need to use a Copy activity in Pipeline1 to ingest data into Container1.
Which two operations can the Copy activity perform? Each correct answer presents a complete solution.
NOTE: Each correct answer is worth one point.

A. insert
B. upsert
C. update
D. delete
E. truncate
Answer: (SHOW ANSWER)
When using a Copy activity in an Azure Data Factory pipeline to ingest data into an Azure Cosmos DB for NoSQL container, the activity is able to execute two distinct write behaviors (operations): Insert and Upsert.
Write Operations Supported
Insert: This operation acts as a standard append. It attempts to add the incoming data as a new document inside the Cosmos DB container. If a document with the exact same unique identifier (id) already exists in the container, the operation will fail with a primary key conflict error.
Upsert: This operation conditionally updates or inserts data. If an incoming document shares the same id as an existing document in the container, the Copy activity will overwrite (update) the existing document. If the id does not match any existing documents, it will create (insert) a brand new document.
Reference:
https://learn.microsoft.com/en-us/fabric/data-factory/connector-azure-cosmosdb-for-nosql-copy-activity

NEW QUESTION: 139
You have an application named App1 that reads the data in an Azure Cosmos DB for NoSQL 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: (SHOW ANSWER)
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

Valid DP-420 Dumps shared by EduDump.com for Helping Passing DP-420 Exam! EduDump.com now offer the **newest DP-420 exam dumps**, the EduDump.com DP-420 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com DP-420 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/DP-420/premium/ (**281** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)