

Microsoft.DP-420-KR.v2026-08-28.q136

□□□□:	DP-420-KR
□□□□:	Designing and Implementing Cloud-Native Applications Using Microsoft Azure Cosmos DB (DP-420 Korean Version)
□□□:	Microsoft
□□ □□ □□□:	136
□□:	v2026-08-28
# □□ □:	126
# □□ □□□:	1360
https://www.krdump.com/Microsoft.DP-420-KR.v2026-08-28.q136.html	

NEW QUESTION: 1

App1□ Azure Cosmos DB Core(SQL API) □□□ □□□ □□□□ □□ □□□□□□□□□. □ □□□□□□□ □□ □□□ □□ □□□ □□□□□. □□□ □□□ □□ □□□ □□□ Eventual□□□. □□□ □□□□ □□ □□ □□□□ □□ □□□ □□□□ □□ □□□□□□. IntegratedCacheItemHitRate□ □ □□□ □□□ □□ □□□□□. IntegratedCacheQueryHitRate □□ 0□□□. □□ □□□□□ □□□□□ □□□□□ □ □□ □□□□□□ □□ □□□ □□□□□□. □□□□□□ App1□ Azure Cosmos DB □□ □□□ □□□□□ □□□□ □□□. □□ □ □□ □□□ □□□□ □□□?

A. □□□□□□ App1□ □□ □□□ □□

B. Azure Cosmos DB □□□□□ □□□ □□

C. Azure Cosmos DB □□□ □□ □□□ □□

D. □□□□□□ 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: 2

□□□ □□

Container1□ Container2□□ □□□ Azure Cosmos for NoSQL □□□□□ □ □ □□□□.

□ □□□□□ □□ □□□ □□□□ □□ □□ □□□ □□□ □□□□□.

- □□□□1: □ □□

The partition key determines how Azure Cosmos DB routes data in partitions. The IoT Device ID is the usual partition key for IoT applications.

Incorrect:

* date

Would be the same during the whole day, which is not good.

Note: Use item ID [Device ID] as the partition key

If your container has a property that has a wide range of possible values, it's likely a great partition key choice. One possible example of such a property is the item ID. For small read-heavy containers or write-heavy containers of any size, the item ID (/id) is naturally a great choice for the partition key.

The system property item ID exists in every item in your container. You might have other properties that represent a logical ID of your item. In many cases, these IDs are also great partition key choices for the same reasons as the item ID.

The item ID is a great partition key choice for the following reasons:

There are a wide range of possible values (one unique item ID per item).

Because there's a unique item ID per item, the item ID does a great job at evenly balancing RU consumption and data storage.

You can easily do efficient point reads since you always know an item's partition key if you know its item ID.

Reference:

<https://learn.microsoft.com/en-us/azure/architecture/solution-ideas/articles/iot-using-cosmos-db>

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

NEW QUESTION: 4

□□□ □□

Container1□□□ □□□ □□□□□ □□□ Azure Cosmos DB for NoSQL □□□ □□□□.

□□□□1□ □□ □□□ □□□ □□ □□□□ □□□□□. □□□□ □□□ □□□ □□□□.

□□ ASC, □□ ASC

□□□ Container1□ □□ □□ □□□ □□□ □□□□□.

□□□□ *

C□□

c.firstName□ "John"□ □□

□□□ c.age > 21

□□□ c._ts < 1612213196

□□ □□□ □□□□□□ □□ □□□□ □□□□ □□□.

□□ □□□ □□□ □□□□ □□□? □□□□□ □□□□□ □□□ □□□ □□□□□□.

□□: □□ □□□ 1□□□□.

Answer Area

Index components:

age

firstName

_ts

ASC,

age

firstName

_ts

DESC

Answer:

Answer Area

Index components:

age

firstName

_ts

ASC,

age

firstName

_ts

DESC

NEW QUESTION: 5

□□□ □□
□□□ □□ Azure Resource Manager(ARM) □□□□ □□□□.

```
{
  "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"
          }
        ]
      ]
    }
  }
}
```

□□□□ □□□ □□□ □□□□□□.

□□ □ □□□ □□, □□□ □□□□ '□'□ □□□□, □□□ □□□ '□□□□'□ □□□□□□.

□□: □□ □□□ 1□□□□.

Answer Area

Statements

When the template is deployed, an Azure Cosmos DB account named acct01 will be created if the account does NOT exist

When the template is deployed, an container named mycontainer in mydb will be created if the container does NOT exist

When the template is deployed, if mycontainer exists and has a partition key on name, the partition key will change to companyid

Yes

No

☐

☐

☐

☐

☐

☐

Answer:

Answer Area

Statements

When the template is deployed, an Azure Cosmos DB account named acct01 will be created if the account does NOT exist

When the template is deployed, an container named mycontainer in mydb will be created if the container does NOT exist

When the template is deployed, if mycontainer exists and has a partition key on name, the partition key will change to companyid

Yes

No

☐

☒

☒

☐

☐

☒

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>

NEW QUESTION: 6

□□□ □□

□□□□□□ □□□□ □ □□ □□□ □□□□ □□ container1□□□ □□□ Azure Cosmos DB □□□□□ □□□□.

Container1 has 100 GB of storage.
- Container1 RU consumption
- HTTP status code 429 requests are 10% of total requests.
What is the percentage of requests that have an HTTP status code of 429?
Options: 10%, 5%, 1%, 0%.

Answer Area

 Microsoft

Normalized RU consumption:

	▼
Below 5%	
At 50%	
Reaching 100%	

The percentage of requests that have an HTTP status code of 429:

	▼
Below 5%	
At 50%	
Reaching 100%	

Answer:
Answer Area

 Microsoft

Normalized RU consumption:

	▼
Below 5%	
At 50%	
Reaching 100%	

The percentage of requests that have an HTTP status code of 429:

	▼
Below 5%	
At 50%	
Reaching 100%	

NEW QUESTION: 7

Container1 has 100 GB of storage.
Azure Cosmos DB for NoSQL Container1 has 100 GB of storage.
Container1 Always Encrypted is enabled.
Always Encrypted Container1 stores JSON data.
What is the percentage of requests that have an HTTP status code of 429?
Options: 10%, 5%, 1%, 0%.

Answer Area

Leaf values will be:

Encrypted

In plain text

Removed

Intermediate sub-property names will be:

Encrypted

In plain text

Removed

Answer:

Answer Area

Leaf values will be:

▼

Encrypted

In plain text

Removed

Intermediate sub-property names will be:

▼

Encrypted

In plain text

Removed

NEW QUESTION: 8

□□□ □□

Azure Cosmos DB for NoSQL □□□□□ □□□□. □ □□□□□□ □□□ □□ □□□ □□ □□□ □□□□.

Property	Data type	Filtered in queries
dateOfBirth	Date	Yes
hasProvidedTaxNumber	Boolean	Yes
healthStatus	String	No

□□□□□ □□□ □□□□ □□□□□ Always Encrypted□ □□□□ □□□. □ □□□ □□ □□ □□□ □□□ □□□□ □□□□ □□□ □□□□ □□□□ □□□. □ □□□ □□□ □ □□ □□ □□□ □□□ □□□ □□□□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

ANSWER AREA

dateOfBirth:	Deterministic Randomized No encryption
healthStatus:	Deterministic Randomized No encryption
hasProvidedTaxNumber:	Deterministic Randomized No encryption

Answer:

Answer Area

dateOfBirth:

healthStatus:

hasProvidedTaxNumber:

NEW QUESTION: 9

□□□ □□ □□□□□ □□□ □□□, □ □□□ 4□□□ □□ □□□ □□□□ □□□, 1, 2□□□□ □□□ □□ □□□ □□, 3, 4□□□□ □□ □□ □□□ □□□□. □ □□□ □□□□ □□ □ □□□ □ □ 50,000 RU/s□ □□□□. □□ □□□□□□ □□ □ □□□□□ □□□ □□ countryId□ □□□ □□ □□□□□. □□ □ □□ □□□ □□□□□?

- A. `countryId`
- B. `countryId`
- C. `countryId`
- D. `countryId`

Answer: B (LEAVE A REPLY)

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

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

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

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

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

Reference:

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

NEW QUESTION: 10

Effective Composite Index Design

To ensure the query is served by the index rather than a scan, your index should follow these rules:

Sequence Matters: The properties in your composite index must appear in the exact order they are used in the WHERE and ORDER BY clauses.

Filter Logic: Include all properties used in your WHERE filters (joined by AND).

Sort Integration: If you have an ORDER BY clause, those properties must come after the equality filters in the index definition.

Order Match: The ascending/descending order in the index definition must match the query for multiple-property sorting.

Example Index Definition:

Property 1: /customerId (Ascending)

Property 2: /orderStatus (Ascending)

Property 3: /orderDate (Descending)

Reference:

<https://oneuptime.com/blog/post/2026-02-02-keyset-pagination/view>

NEW QUESTION: 12

□□□ □□

Azure Cosmos DB for NoSQL □□□□ □□□□ □□□□.

□□□ □□ □□ □□□ □□ □□□ □□ □□ □□ □□□□ □□□ Container1□□□ □ □□□□□ □□□ □□□□□.

Column name	Data type	Data length in bytes
Car Brand	String	30
Car Model	String	30
Model Year	Int	4
Part Name	String	50
Short Description	String	500
Storage Location	String	50

□□□□ □□ □□ □□□ □□ Container1□ □□□□□.

Car Brand	Car Model	Model Year	Part Name	Storage Location	Short Description
Ford	Escort	1999	Door Handle	Shelf 5, Bin 3	Description1
Ford	Escort	2000	Door Handle	Shelf 5, Bin 5	Description2
Ford	Escort	2001	Door Handle	Shelf 32, Bin 17	Description3
Ford	Escape	1999	Door Handle	Shelf 43, Bin 12	Description4
Chevrolet	Pacer	1975	Door Handle	Shelf 19, Bin 12	Description5
Chevrolet	Bolt	2024	Door Release	Shelf 20, Bin 17	Description6

□□□□□□□ □□□ □□□ □□□□ □□□□ □□□ □□□. □ □□□□ I/O □□□ □□□□□ □□ □□ □□□ □□□□□ □□□.

□□□ □□□ □□ □□□ □□□ □□ □□ □□ □□□□ □□ □□□□?

□□□□□ □□□□□ □□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

Answer Area

Physical partition key:

Car Brand
Car Model
Model Year
Part Name
Storage Location

Logical partition key:

Car Brand
Car Model
Model Year
Part Name
Storage Location

Answer:
Answer Area

Physical partition key:

Car Brand
Car Model
Model Year
Part Name
Storage Location

Logical partition key:

Car Brand
Car Model
Model Year
Part Name
Storage Location

NEW QUESTION: 13

account1 Azure Cosmos DB for NoSQL . account1 Container1 . App1 Container1 .

Azure Monitor□ □□□ account1□ □□ □ □□□ □□□□.

□□□□□ □□□□ □□□□ □□□ □ □□□□ □□ □□□ □□□□□ □□□□□, App1□ □□□□ HTTP □□□ □□□□□.

429 .

□□ □□□ □□□□ □□□□ □□ □□□□ □□□□ □□□□□ □□□□ □□□.

Azure Cosmos DB Insights□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

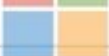
Answer Area

Metric:

Split chart by:

Answer:

Answer Area

 Microsoft

Metric:

Metadata Requests That Exceeded Capacity from System

Normalized RU Consumption (%) By PartitionKeyRangeID from Throughput

Total Requests By Status Code from Requests

Split chart by:

DatabaseName

Operation Type

PartitionKeyRangeId

NEW QUESTION: 14

Container1 Azure Cosmos DB for NoSQL Pipeline1 Azure Data Factory Azure

Container1□ □□□□ □□□□ Pipeline1□□ Copy □□□ □□□□ □□□.

Copy □□□ □□□ □ □□ □ □□ □□□ □□□□□? □ □□□ □□□ □□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

A. ☐ ☐

B. □□□

C. ☐ ☐ ☐ ☐

D. ☐

E. ☐

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: 15

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □ □□□□ □□□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□□ □ □ □□□ □ □□□, □□□□ □□ □□ □□□ □□□□.

□ □□□□ □□□ □□□ □□□ □□ □□□□ □□□ □ □□□□. □□□ □ □□□□ □□ □□□ □□□□ □□□□.

□□ □□ □□□□ □□□□ account1□□□□ Azure Cosmos DB for NoSQL □□□□ □□□□.

account1□ □□□□□□ □□ □□ □□□□ □□ □□□ □□ □□ □□□ □ Azure □□□□ □□□□ □□□.

□□ □□: □□ □□□□ Event Grid □□□□ □□□□ Azure Event Grid□ □□□ □□□□ □□□□ □□□□□.

□□□□ □□□ □□□□□?

A. ☐

B. ☐

Answer: B (LEAVE A REPLY)

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: 16

□□ □□ 1 - □□□□ □□□□

□□

Litware, Inc.□ □□□ □□□ □ □□□ □□□□□□□□. Litware□ □□□□ □□□ □□ □□□□□□□ □□ □□□□. □□ □□□□ 50□□ □□□□ □□□□ □□□ □□□ □□ □□□□ □□□□. □ □□□ □□ □□ □□ □□□□ □□□□. □□ □□.

□□□□ □ □□□ □□□ □□. Litware□ □□ □□ □□□ □□□□ □□□□ Azure □□□□ □□□□ □□□□.

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.

productdb is a database in the account1. con-product is a container in the productdb database. con-productVendor is a container in the productdb database. con-product is a container in the productdb database.

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

con-productVendor is a container in the productdb database. SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Existing environment.

Litware is a company that is using Azure IoT Hub.

- con-productVendor is a container in the productdb database. con-product is a container in the productdb database.

- 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.

- 429 is a number that is used in the IoT hub.

is a number that is used in the IoT hub.

is a number that is used in the IoT hub.

Litware is a company that is using Azure Cosmos DB for NoSQL. The account1 is a database in the account1. The account2 is a database in the account2. The account3 is a database in the account3.

Litware is a company that is using Azure Cosmos DB for NoSQL.

- The account1 is a database in the account1.

- account1 is a database in the account1.

- con-product is a container in the productdb database.

- Azure Functions is a service that is used in the IoT hub.

- The account1 is a database in the account1.

- App2 is a database in the account2.

- App2 is a database in the account2.

is a number that is used in the IoT hub.

is a number that is used in the IoT hub.

Litware is a company that is using Azure Cosmos DB for NoSQL.

The account1 is a database in the account1. The account2 is a database in the account2. The account3 is a database in the account3.

- IoT is a service that is used in the IoT hub.

- The account1 is a database in the account1.

is a number that is used in the IoT hub.

is a number that is used in the IoT hub.

Litware is a company that is using Azure Cosmos DB for NoSQL.

- The account1 is a database in the account1.

In Azure AI Search, the prefix query type allows you to perform "starts-with" matching using the Simple Query Parser.

Native Support: It is supported by the default simple query parser, meaning you do not need to change the queryType property in your REST API call.

Simple Syntax: You simply append a * to a partial term (e.g., search=cap*).

No Custom Configuration: Unlike edge n-gram tokenization, which requires you to manually define and configure custom analyzers and field mappings in your index schema, prefix search works out-of-the-box on standard searchable fields.

[C]

The other query type that can also be used to retrieve text data while keeping coding to a minimum is wildcard.

In Azure AI Search, wildcard queries (using * and ?) are supported in the Simple Query Syntax, which is the default query language. This allows you to perform basic pattern matching directly from user input without needing to set advanced parameters or implement complex custom analyzers.

Reference:

<https://medium.com/@bryanhanner/building-search-into-your-application-84943cd359d4>

<https://docs.azure.cn/en-us/search/search-features-list>

NEW QUESTION: 18

□□□ □□

□□□ Container1□□□ □□□ □□□□□ □□□□□ Azure Cosmos DB for NoSQL □□□ □□□□ □□□□.

Function1□□□ □□□ Azure Function□ □□□□ □□□□.

Container1□□ □□□ □□□□□ □□□□□ □ Function1□ □□□□□ □□ □□□.

□ □□□ □□□ □□□□ □□□? □□□□ □□□□□ □□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□□.

Answer Area

```
{
  "type": "cosmosDBTrigger",
  "name": "changes",
  "direction": 

|          |   |
|----------|---|
|          | ▼ |
| "change" |   |
| "in"     |   |
| "out"    |   |
| "update" |   |

 ,
  "connectionStringSetting": "myconnectionstring",
  "databaseName": "products",
  "collectionName": "cars",
  

|                       |   |
|-----------------------|---|
|                       | ▼ |
| "binding"             |   |
| "leaseCollectionName" |   |
| "partitionKey"        |   |
| "sqlQuery"            |   |

 : "feed",
  "createLeaseCollectionIfNotExists": false
}
```



Answer:

Answer Area

```
{
  "type": "cosmosDBTrigger",
  "name": "changes",
  "direction": 

"change"



"in"



"out"



"update"


    ,
  "connectionStringSetting": "myconnectionstring",
  "databaseName": "products",
  "collectionName": "cars",
  

"binding"



"leaseCollectionName"



"partitionKey"



"sqlQuery"


    : "feed",

  "createLeaseCollectionIfNotExists": false
}
```



Which of the following SQL queries will return the correct result?

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

The correct result is a table with 5 rows and 3 columns. The first column is Name, the second column is Type, and the third column is Orders. The rows are ordered by the number of orders in descending order.

```
{
  "name": "apple",
  "type": ["fruit", "exotic"],
  "orders": 10000
}
```

Which of the following SQL queries will return the correct result? (Select all that apply.)

A. `SELECT TOP 5 i.name, i.types, i.orders FROM items i WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit') ORDER BY i.orders, i.types`

B. `SELECT TOP 5 i.name, i.types, i.orders FROM items i WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit') ORDER BY i.orders DESC`

C. `SELECT TOP 5 i.name, i.types, i.orders FROM items i WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit') ORDER BY i.types DESC`

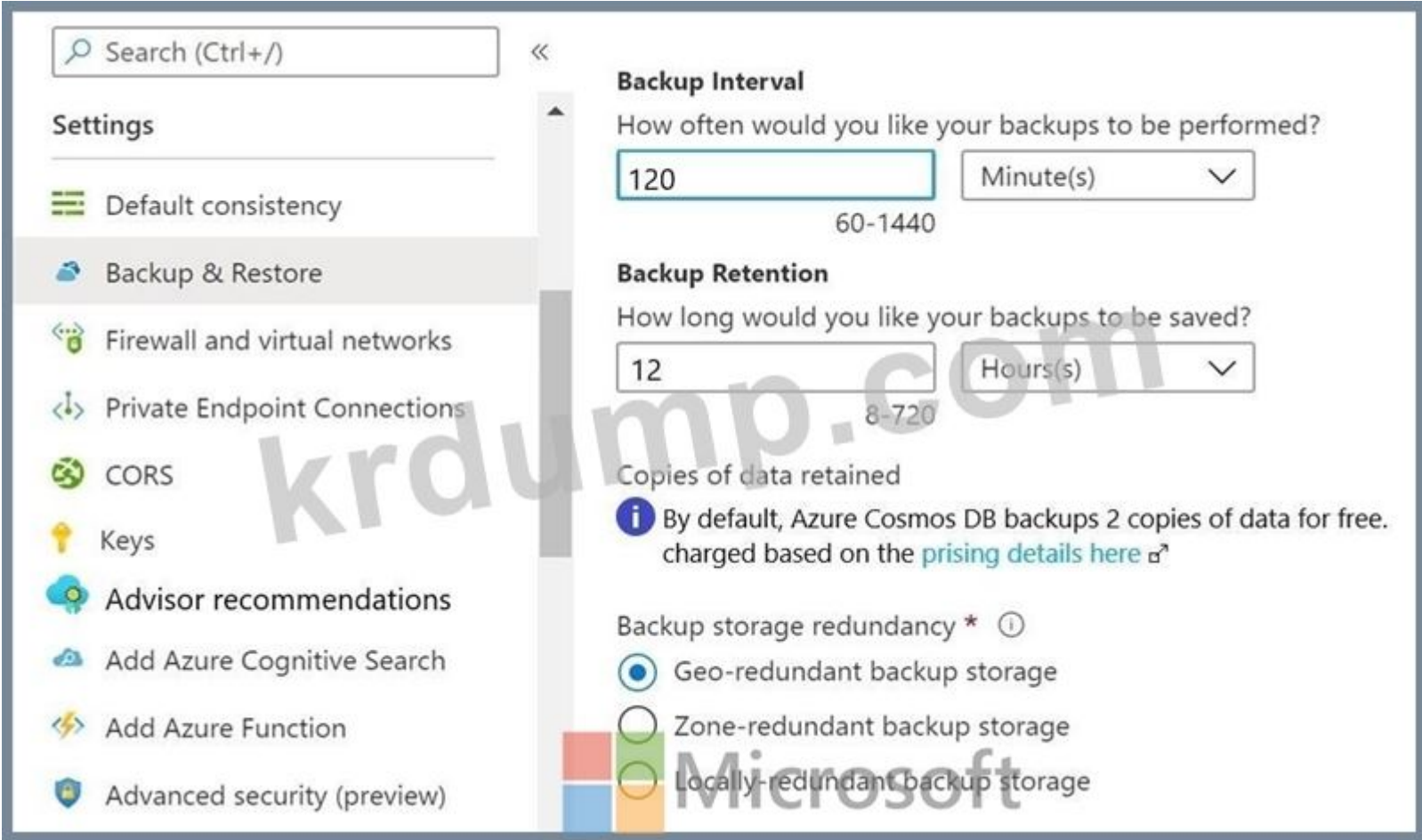
D. `SELECT TOP 5 i.name, i.types, i.orders FROM items i WHERE ARRAY_CONTAINS(i.types, {name: 'fruit'}) ORDER BY i.orders DESC`

Answer: B,D (LEAVE A REPLY)

ARRAY_CONTAINS returns a Boolean indicating whether the array contains the specified value.
You can check for a partial or full match of an object by using a boolean expression within the command.
Incorrect Answers:
A: Default sorting ordering is Ascending. Must use Descending order.
C: Order on Orders not on Type.
Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/sql-query-array-contains>

NEW QUESTION: 20

☐☐☐ ☐☐
☐☐ ☐☐☐ ☐☐ Azure Cosmos DB for NoSQL ☐☐☐ ☐☐ ☐☐☐ ☐☐☐☐☐.



☐☐ ☐☐☐ ☐☐☐ ☐☐☐ ☐☐☐☐ ☐ ☐☐☐ ☐☐☐☐ ☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐☐.
☐☐: ☐☐ ☐☐☐ 1☐☐☐☐.

Answer Area

The current backup interval and retention policy provide protection for [answer choice]

	▼
1 hour	
2 hours	
6 hours	
12 hours	
3 days	

In case of emergency, you must [answer choice] to restore the backup

	▼
create a support ticket	
use the Backup & Restore settings	
use the Point in Time Restore settings	

Answer:

Answer Area

The current backup interval and retention policy provide protection for [answer choice]

▼

1 hour

2 hours

6 hours

12 hours

3 days

In case of emergency, you must [answer choice] to restore the backup

▼

create a support ticket

use the Backup & Restore settings

use the Point in Time Restore settings

Explanation:

Box 1: 12 hours

The number of copies stored depends on the interval and the retention period.

The default retention backup intervall is 240 minutes (4h) and the retention period is 8h, thats why

2 copies are stored by default. The first 2 copies are free as well.

In this example, every 2h a copy is created and is kept for 12h. This means there can be up to 6 copies. 4 of the 6 copies are billed according to Azure blob storage pricing.

Box 2: create a support ticket

Request data restore from a backup

If you accidentally delete your database or a container, you can file a support ticket or call the Azure support to restore the data from automatic online backups.

Note: With Azure Cosmos DB SQL API accounts, you can also maintain your own backups by using one of the following approaches:

Use Azure Data Factory to move data periodically to a storage of your choice.

Use Azure Cosmos DB change feed to read data periodically for full backups or for incremental changes, and store it in your own storage.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/configure-periodic-backup-restore>

NEW QUESTION: 21

□□ □□ □□□ □□ Azure Cosmos DB for NoSQL □□□ □□□□.

□□□□□□□ □□□□ □□ □□□ □□ □□ □□□ □□ □□(RU/s)□ □□□ □□ □□□ □□□ □□□.

□□ Azure Monitor □□□ □□□□ □□□?

A. □□ □

B. □□ □□□

C. □□ □□□

D. □ □□ □□

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 22

□□□ □□

Azure Cosmos DB for NoSQL □□□□□□(DB1)□ □□□, □ □□□□□□□ Container1□□□ □□□□□ □□□□□□. Container1□ □□□ □□ /city□□□.

□□ □□□ DB1□ □□□□ App1□□□□ □□□□ □□□□ □□ □□□□ □□□□.

App1□ Container1□ □□□ □□□□□□□ □□ □□□□ □□ □□□ □□□□□.

App1□ □□ □□ □□□ □□ □□□□ □□□.

- □□□ ETag □□ □□□ □□ □□ □□□ □□□□□ □□□.

□□ □□□□□ □□ □□□ □□□□□□□□.

- □□ □□□ □□ □□□ □□ □□□□ □□□□□ □□□□□□.

□□ □□.

C# □□□ □□□ □□□□ □□□? □□□□ □□ □□□□ □□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

Answer Area

```
public class Program
{
    public static async Task Main(string[] args)
    {

```

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

```
{ IfMatchEtag = "etag1" };
```

```
{ IfNoneMatchEtag = "etag1" };
```

```
{ IfNoneMatchEtag = "etag1", EnableContentResponseOnWrite = false };
```

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
            {
                PartitionKey = new PartitionKey(batch.UpsertItem(new { id = "1", city = "London" }));
            });
            // Ensure rollback if ETag value changes
            container.CreateTransactionalBatch(new PartitionKey
            {
                city = "London"
            });
        }
    }
}
```

```
( IfMatchEtag = "etag1" ));  
( IfNoneMatchEtag = "etag1" ));  
( IfNoneMatchEtag = "etag1", EnableContentResponseOnWrite = false ));
```

```
["city");  
["/city");  
["London");
```

NEW QUESTION: 23

□□□ □□
 Azure Cosmos DB for NoSQL □□□ □□□□□□□ □□□□ □□□□. □ □□□□□□□ □□□□□ □□□ □□□□ □□□ □ □□□ □□ □□□□□□□□ □□□ □□□□□□. □□□□□ □□ □□□□ □□□□
 □□□ □□ □□ □□□□.
 □□ □□ □□ □□ □□□□ □□□□ □□□□.

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

□□ □□□□□□ □□ □□ □□ □□ □□□□.
 - □□□□ □□□□ □□□□ □□ □ □□□□.
 - □□□ □ □□ □□□□ □□ □□ □□ □□□□ □□ □□□□.
 1,000.
 - □□□□□ □□ □□□□ □□□□□ □□□ □ □ □□□□.
 □□□□ □□ □□ □□ 100□□□□, 1,000□□ □□ □□□□ □□ □□□□. □□□□ □□ 20□□ □□□□ □□□ □ □□□□. □□ □ □□□ □□, □□□ □□□□ '□'□ □□□□, □□□ □□□ '□□□□'□ □□□□ □.
 □□: □□ □□□ 1□□□□.

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	☐	☐

Answer:

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	☐	☒

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: 24

Q1 - Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q20 Q21 Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 Q33 Q34 Q35 Q36 Q37 Q38 Q39 Q40 Q41 Q42 Q43 Q44 Q45 Q46 Q47 Q48 Q49 Q50 Q51 Q52 Q53 Q54 Q55 Q56 Q57 Q58 Q59 Q60 Q61 Q62 Q63 Q64 Q65 Q66 Q67 Q68 Q69 Q70 Q71 Q72 Q73 Q74 Q75 Q76 Q77 Q78 Q79 Q80 Q81 Q82 Q83 Q84 Q85 Q86 Q87 Q88 Q89 Q90 Q91 Q92 Q93 Q94 Q95 Q96 Q97 Q98 Q99 Q100

Q101 Q102 Q103 Q104 Q105 Q106 Q107 Q108 Q109 Q110 Q111 Q112 Q113 Q114 Q115 Q116 Q117 Q118 Q119 Q120 Q121 Q122 Q123 Q124 Q125 Q126 Q127 Q128 Q129 Q130 Q131 Q132 Q133 Q134 Q135 Q136 Q137 Q138 Q139 Q140 Q141 Q142 Q143 Q144 Q145 Q146 Q147 Q148 Q149 Q150 Q151 Q152 Q153 Q154 Q155 Q156 Q157 Q158 Q159 Q160 Q161 Q162 Q163 Q164 Q165 Q166 Q167 Q168 Q169 Q170 Q171 Q172 Q173 Q174 Q175 Q176 Q177 Q178 Q179 Q180 Q181 Q182 Q183 Q184 Q185 Q186 Q187 Q188 Q189 Q190 Q191 Q192 Q193 Q194 Q195 Q196 Q197 Q198 Q199 Q200

Q201 Q202 Q203 Q204 Q205 Q206 Q207 Q208 Q209 Q210 Q211 Q212 Q213 Q214 Q215 Q216 Q217 Q218 Q219 Q220 Q221 Q222 Q223 Q224 Q225 Q226 Q227 Q228 Q229 Q230 Q231 Q232 Q233 Q234 Q235 Q236 Q237 Q238 Q239 Q240 Q241 Q242 Q243 Q244 Q245 Q246 Q247 Q248 Q249 Q250 Q251 Q252 Q253 Q254 Q255 Q256 Q257 Q258 Q259 Q260 Q261 Q262 Q263 Q264 Q265 Q266 Q267 Q268 Q269 Q270 Q271 Q272 Q273 Q274 Q275 Q276 Q277 Q278 Q279 Q280 Q281 Q282 Q283 Q284 Q285 Q286 Q287 Q288 Q289 Q290 Q291 Q292 Q293 Q294 Q295 Q296 Q297 Q298 Q299 Q300

Q301 Q302 Q303 Q304 Q305 Q306 Q307 Q308 Q309 Q310 Q311 Q312 Q313 Q314 Q315 Q316 Q317 Q318 Q319 Q320 Q321 Q322 Q323 Q324 Q325 Q326 Q327 Q328 Q329 Q330 Q331 Q332 Q333 Q334 Q335 Q336 Q337 Q338 Q339 Q340 Q341 Q342 Q343 Q344 Q345 Q346 Q347 Q348 Q349 Q350 Q351 Q352 Q353 Q354 Q355 Q356 Q357 Q358 Q359 Q360 Q361 Q362 Q363 Q364 Q365 Q366 Q367 Q368 Q369 Q370 Q371 Q372 Q373 Q374 Q375 Q376 Q377 Q378 Q379 Q380 Q381 Q382 Q383 Q384 Q385 Q386 Q387 Q388 Q389 Q390 Q391 Q392 Q393 Q394 Q395 Q396 Q397 Q398 Q399 Q400

Q401 Q402 Q403 Q404 Q405 Q406 Q407 Q408 Q409 Q410 Q411 Q412 Q413 Q414 Q415 Q416 Q417 Q418 Q419 Q420 Q421 Q422 Q423 Q424 Q425 Q426 Q427 Q428 Q429 Q430 Q431 Q432 Q433 Q434 Q435 Q436 Q437 Q438 Q439 Q440 Q441 Q442 Q443 Q444 Q445 Q446 Q447 Q448 Q449 Q450 Q451 Q452 Q453 Q454 Q455 Q456 Q457 Q458 Q459 Q460 Q461 Q462 Q463 Q464 Q465 Q466 Q467 Q468 Q469 Q470 Q471 Q472 Q473 Q474 Q475 Q476 Q477 Q478 Q479 Q480 Q481 Q482 Q483 Q484 Q485 Q486 Q487 Q488 Q489 Q490 Q491 Q492 Q493 Q494 Q495 Q496 Q497 Q498 Q499 Q500

Q501 Q502 Q503 Q504 Q505 Q506 Q507 Q508 Q509 Q510 Q511 Q512 Q513 Q514 Q515 Q516 Q517 Q518 Q519 Q520 Q521 Q522 Q523 Q524 Q525 Q526 Q527 Q528 Q529 Q530 Q531 Q532 Q533 Q534 Q535 Q536 Q537 Q538 Q539 Q540 Q541 Q542 Q543 Q544 Q545 Q546 Q547 Q548 Q549 Q550 Q551 Q552 Q553 Q554 Q555 Q556 Q557 Q558 Q559 Q560 Q561 Q562 Q563 Q564 Q565 Q566 Q567 Q568 Q569 Q570 Q571 Q572 Q573 Q574 Q575 Q576 Q577 Q578 Q579 Q580 Q581 Q582 Q583 Q584 Q585 Q586 Q587 Q588 Q589 Q590 Q591 Q592 Q593 Q594 Q595 Q596 Q597 Q598 Q599 Q600

Q601 Q602 Q603 Q604 Q605 Q606 Q607 Q608 Q609 Q610 Q611 Q612 Q613 Q614 Q615 Q616 Q617 Q618 Q619 Q620 Q621 Q622 Q623 Q624 Q625 Q626 Q627 Q628 Q629 Q630 Q631 Q632 Q633 Q634 Q635 Q636 Q637 Q638 Q639 Q640 Q641 Q642 Q643 Q644 Q645 Q646 Q647 Q648 Q649 Q650 Q651 Q652 Q653 Q654 Q655 Q656 Q657 Q658 Q659 Q660 Q661 Q662 Q663 Q664 Q665 Q666 Q667 Q668 Q669 Q670 Q671 Q672 Q673 Q674 Q675 Q676 Q677 Q678 Q679 Q680 Q681 Q682 Q683 Q684 Q685 Q686 Q687 Q688 Q689 Q690 Q691 Q692 Q693 Q694 Q695 Q696 Q697 Q698 Q699 Q700

Q701 Q702 Q703 Q704 Q705 Q706 Q707 Q708 Q709 Q710 Q711 Q712 Q713 Q714 Q715 Q716 Q717 Q718 Q719 Q720 Q721 Q722 Q723 Q724 Q725 Q726 Q727 Q728 Q729 Q730 Q731 Q732 Q733 Q734 Q735 Q736 Q737 Q738 Q739 Q740 Q741 Q742 Q743 Q744 Q745 Q746 Q747 Q748 Q749 Q750 Q751 Q752 Q753 Q754 Q755 Q756 Q757 Q758 Q759 Q760 Q761 Q762 Q763 Q764 Q765 Q766 Q767 Q768 Q769 Q770 Q771 Q772 Q773 Q774 Q775 Q776 Q777 Q778 Q779 Q780 Q781 Q782 Q783 Q784 Q785 Q786 Q787 Q788 Q789 Q790 Q791 Q792 Q793 Q794 Q795 Q796 Q797 Q798 Q799 Q800

Q801 Q802 Q803 Q804 Q805 Q806 Q807 Q808 Q809 Q810 Q811 Q812 Q813 Q814 Q815 Q816 Q817 Q818 Q819 Q820 Q821 Q822 Q823 Q824 Q825 Q826 Q827 Q828 Q829 Q830 Q831 Q832 Q833 Q834 Q835 Q836 Q837 Q838 Q839 Q840 Q841 Q842 Q843 Q844 Q845 Q846 Q847 Q848 Q849 Q850 Q851 Q852 Q853 Q854 Q855 Q856 Q857 Q858 Q859 Q860 Q861 Q862 Q863 Q864 Q865 Q866 Q867 Q868 Q869 Q870 Q871 Q872 Q873 Q874 Q875 Q876 Q877 Q878 Q879 Q880 Q881 Q882 Q883 Q884 Q885 Q886 Q887 Q888 Q889 Q890 Q891 Q892 Q893 Q894 Q895 Q896 Q897 Q898 Q899 Q900

Q901 Q902 Q903 Q904 Q905 Q906 Q907 Q908 Q909 Q910 Q911 Q912 Q913 Q914 Q915 Q916 Q917 Q918 Q919 Q920 Q921 Q922 Q923 Q924 Q925 Q926 Q927 Q928 Q929 Q930 Q931 Q932 Q933 Q934 Q935 Q936 Q937 Q938 Q939 Q940 Q941 Q942 Q943 Q944 Q945 Q946 Q947 Q948 Q949 Q950 Q951 Q952 Q953 Q954 Q955 Q956 Q957 Q958 Q959 Q960 Q961 Q962 Q963 Q964 Q965 Q966 Q967 Q968 Q969 Q970 Q971 Q972 Q973 Q974 Q975 Q976 Q977 Q978 Q979 Q980 Q981 Q982 Q983 Q984 Q985 Q986 Q987 Q988 Q989 Q990 Q991 Q992 Q993 Q994 Q995 Q996 Q997 Q998 Q999 Q1000

Q1001 Q1002 Q1003 Q1004 Q1005 Q1006 Q1007 Q1008 Q1009 Q1010 Q1011 Q1012 Q1013 Q1014 Q1015 Q1016 Q1017 Q1018 Q1019 Q1020 Q1021 Q1022 Q1023 Q1024 Q1025 Q1026 Q1027 Q1028 Q1029 Q1030 Q1031 Q1032 Q1033 Q1034 Q1035 Q1036 Q1037 Q1038 Q1039 Q1040 Q1041 Q1042 Q1043 Q1044 Q1045 Q1046 Q1047 Q1048 Q1049 Q1050 Q1051 Q1052 Q1053 Q1054 Q1055 Q1056 Q1057 Q1058 Q1059 Q1060 Q1061 Q1062 Q1063 Q1064 Q1065 Q1066 Q1067 Q1068 Q1069 Q1070 Q1071 Q1072 Q1073 Q1074 Q1075 Q1076 Q1077 Q1078 Q1079 Q1080 Q1081 Q1082 Q1083 Q1084 Q1085 Q1086 Q1087 Q1088 Q1089 Q1090 Q1091 Q1092 Q1093 Q1094 Q1095 Q1096 Q1097 Q1098 Q1099 Q1100

Q1101 Q1102 Q1103 Q1104 Q1105 Q1106 Q1107 Q1108 Q1109 Q1110 Q1111 Q1112 Q1113 Q1114 Q1115 Q1116 Q1117 Q1118 Q1119 Q1120 Q1121 Q1122 Q1123 Q1124 Q1125 Q1126 Q1127 Q1128 Q1129 Q1130 Q1131 Q1132 Q1133 Q1134 Q1135 Q1136 Q1137 Q1138 Q1139 Q1140 Q1141 Q1142 Q1143 Q1144 Q1145 Q1146 Q1147 Q1148 Q1149 Q1150 Q1151 Q1152 Q1153 Q1154 Q1155 Q1156 Q1157 Q1158 Q1159 Q1160 Q1161 Q1162 Q1163 Q1164 Q1165 Q1166 Q1167 Q1168 Q1169 Q1170 Q1171 Q1172 Q1173 Q1174 Q1175 Q1176 Q1177 Q1178 Q1179 Q1180 Q1181 Q1182 Q1183 Q1184 Q1185 Q1186 Q1187 Q1188 Q1189 Q1190 Q1191 Q1192 Q1193 Q1194 Q1195 Q1196 Q1197 Q1198 Q1199 Q1200

Q1201 Q1202 Q1203 Q1204 Q1205 Q1206 Q1207 Q1208 Q1209 Q1210 Q1211 Q1212 Q1213 Q1214 Q1215 Q1216 Q1217 Q1218 Q1219 Q1220 Q1221 Q1222 Q1223 Q1224 Q1225 Q1226 Q1227 Q1228 Q1229 Q1230 Q1231 Q1232 Q1233 Q1234 Q1235 Q1236 Q1237 Q1238 Q1239 Q1240 Q1241 Q1242 Q1243 Q1244 Q1245 Q1246 Q1247 Q1248 Q1249 Q1250 Q1251 Q1252 Q1253 Q1254 Q1255 Q1256 Q1257 Q1258 Q1259 Q1260 Q1261 Q1262 Q1263 Q1264 Q1265 Q1266 Q1267 Q1268 Q1269 Q1270 Q1271 Q1272 Q1273 Q1274 Q1275 Q1276 Q1277 Q1278 Q1279 Q1280 Q1281 Q1282 Q1283 Q1284 Q1285 Q1286 Q1287 Q1288 Q1289 Q1290 Q1291 Q1292 Q1293 Q1294 Q1295 Q1296 Q1297 Q1298 Q1299 Q1300

Q1301 Q1302 Q1303 Q1304 Q1305 Q1306 Q1307 Q1308 Q1309 Q1310 Q1311 Q1312 Q1313 Q1314 Q1315 Q1316 Q1317 Q1318 Q1319 Q1320 Q1321 Q1322 Q1323 Q1324 Q1325 Q1326 Q1327 Q1328 Q1329 Q1330 Q1331 Q1332 Q1333 Q1334 Q1335 Q1336 Q1337 Q1338 Q1339 Q1340 Q1341 Q1342 Q1343 Q1344 Q1345 Q1346 Q1347 Q1348 Q1349 Q1350 Q1351 Q1352 Q1353 Q1354 Q1355 Q1356 Q1357 Q1358 Q1359 Q1360 Q1361 Q1362 Q1363 Q1364 Q1365 Q1366 Q1367 Q1368 Q1369 Q1370 Q1371 Q1372 Q1373 Q1374 Q1375 Q1376 Q1377 Q1378 Q1379 Q1380 Q1381 Q1382 Q1383 Q1384 Q1385 Q1386 Q1387 Q1388 Q1389 Q1390 Q1391 Q1392 Q1393 Q1394 Q1395 Q1396 Q1397 Q1398 Q1399 Q1400

Q1401 Q1402 Q1403 Q1404 Q1405 Q1406 Q1407 Q1408 Q1409 Q1410 Q1411 Q1412 Q1413 Q1414 Q1415 Q1416 Q1417 Q1418 Q1419 Q1420 Q1421 Q1422 Q1423 Q1424 Q1425 Q1426 Q1427 Q1428 Q1429 Q1430 Q1431 Q1432 Q1433 Q1434 Q1435 Q1436 Q1437 Q1438 Q1439 Q1440 Q1441 Q1442 Q1443 Q1444 Q1445 Q1446 Q1447 Q1448 Q1449 Q1450 Q1451 Q1452 Q1453 Q1454 Q1455 Q1456 Q1457 Q1458 Q1459 Q1460 Q1461 Q1462 Q1463 Q1464 Q1465 Q1466 Q1467 Q1468 Q1469 Q1470 Q1471 Q1472 Q1473 Q1474 Q1475 Q1476 Q1477 Q1478 Q1479 Q1480 Q1481 Q1482 Q1483 Q1484 Q1485 Q1486 Q1487 Q1488 Q1489 Q1490 Q1491 Q1492 Q1493 Q1494 Q1495 Q1496 Q1497 Q1498 Q1499 Q1500

Q1501 Q1502 Q1503 Q1504 Q1505 Q1506 Q1507 Q1508 Q1509 Q1510 Q1511 Q1512 Q1513 Q1514 Q1515 Q1516 Q1517 Q1518 Q1519 Q1520 Q1521 Q1522 Q1523 Q1524 Q1525 Q1526 Q1527 Q1528 Q1529 Q1530 Q1531 Q1532 Q1533 Q1534 Q1535 Q1536 Q1537 Q1538 Q1539 Q1540 Q1541 Q1542 Q1543 Q1544 Q1545 Q1546 Q1547 Q1548 Q1549 Q1550 Q1551 Q1552 Q1553 Q1554 Q1555 Q1556 Q1557 Q1558 Q1559 Q1560 Q1561 Q1562 Q1563 Q1564 Q1565 Q1566 Q1567 Q1568 Q1569 Q1570 Q1571 Q1572 Q1573 Q1574 Q1575 Q1576 Q1577 Q1578 Q1579 Q1580 Q1581 Q1582 Q1583 Q1584 Q1585 Q1586 Q1587 Q1588 Q1589 Q1590 Q1591 Q1592 Q1593 Q1594 Q1595 Q1596 Q1597 Q1598 Q1599 Q1600

Q1601 Q1602 Q1603 Q1604 Q1605 Q1606 Q1607 Q1608 Q1609 Q1610 Q1611 Q1612 Q1613 Q1614 Q1615 Q1616 Q1617 Q1618 Q1619 Q1620 Q1621 Q1622 Q1623 Q1624 Q1625 Q1626 Q1627 Q1628 Q1629 Q1630 Q1631 Q1632 Q1633 Q1634 Q1635 Q1636 Q1637 Q1638 Q1639 Q1640 Q1641 Q1642 Q1643 Q1644 Q1645 Q1646 Q1647 Q1648 Q1649 Q1650 Q1651 Q1652 Q1653 Q1654 Q1655 Q1656 Q1657 Q1658 Q1659 Q1660 Q1661 Q1662 Q1663 Q1664 Q1665 Q1666 Q1667 Q1668 Q1669 Q1670 Q1671 Q1672 Q1673 Q1674 Q1675 Q1676 Q1677 Q1678 Q1679 Q1680 Q1681 Q1682 Q1683 Q1684 Q1685 Q1686 Q1687 Q1688 Q1689 Q1690 Q1691 Q1692 Q1693 Q1694 Q1695 Q1696 Q1697 Q1698 Q1699 Q1700

Q1701 Q1702 Q1703 Q1704 Q1705 Q1706 Q1707 Q1708 Q1709 Q1710 Q1711 Q1712 Q1713 Q1714 Q1715 Q1716 Q1717 Q1718 Q1719 Q1720 Q1721 Q1722 Q1723 Q1724 Q1725 Q1726 Q1727 Q1728 Q1729 Q1730 Q1731 Q1732 Q1733 Q1734 Q1735 Q1736 Q1737 Q1738 Q1739 Q1740 Q1741 Q1742 Q1743 Q1744 Q1745 Q1746 Q1747 Q1748 Q1749 Q1750 Q1751 Q1752 Q1753 Q1754 Q1755 Q1756 Q1757 Q1758 Q1759 Q1760 Q1761 Q1762 Q1763 Q1764 Q1765 Q1766 Q1767 Q1768 Q1769 Q1770 Q1771 Q1772 Q1773 Q1774 Q1775 Q1776 Q1777 Q1778 Q1779 Q1780 Q1781 Q1782 Q1783 Q1784 Q1785 Q1786 Q1787 Q1788 Q1789 Q1790 Q1791 Q1792 Q1793 Q1794 Q1795 Q1796 Q1797 Q1798 Q1799 Q1800

Q1801 Q1802 Q1803 Q1804 Q1805 Q1806 Q1807 Q1808 Q1809 Q1810 Q1811 Q1812 Q1813 Q1814 Q1815 Q1816 Q1817 Q1818 Q1819 Q1820 Q1821 Q1822 Q1823 Q1824 Q1825 Q1826 Q1827 Q1828 Q1829 Q1830 Q1831 Q1832 Q1833 Q1834 Q1835 Q1836 Q1837 Q1838 Q1839 Q1840 Q1841 Q1842 Q1843 Q1844 Q1845 Q1846 Q1847 Q1848 Q1849 Q1850 Q1851 Q1852 Q1853 Q1854 Q1855 Q1856 Q1857 Q1858 Q1859 Q1860 Q1861 Q1862 Q1863 Q1864 Q1865 Q1866 Q1867 Q1868 Q1869 Q1870 Q1871 Q1872 Q1873 Q1874 Q1875 Q1876 Q1877 Q1878 Q1879 Q1880 Q1881 Q1882 Q1883 Q1884 Q1885 Q1886 Q1887 Q1888 Q1889 Q1890 Q1891 Q1892 Q1893 Q1894 Q1895 Q1896 Q1897 Q1898 Q1899 Q1900

Q1901 Q1902 Q1903 Q1904 Q1905 Q1906 Q1907 Q1908 Q1909 Q1910 Q1911 Q1912 Q1913 Q1914 Q1915 Q1916 Q1917 Q1918 Q1919 Q1920 Q1921 Q1922 Q1923 Q1924 Q1925 Q1926 Q1927 Q1928 Q1929 Q1930 Q1931 Q1932 Q1933 Q1934 Q1935 Q1936 Q1937 Q1938 Q1939 Q1940 Q1941 Q1942 Q1943 Q1944 Q1945 Q1946 Q1947 Q1948 Q1949 Q1950 Q1951 Q1952 Q1953 Q1954 Q1955 Q1956 Q1957 Q1958 Q1959 Q1960 Q1961 Q1962 Q1963 Q1964 Q1965 Q1966 Q1967 Q1968 Q1969 Q1970 Q1971 Q1972 Q1973 Q1974 Q1975 Q1976 Q1977 Q1978 Q1979 Q1980 Q1981 Q1982 Q1983 Q1984 Q1985 Q1986 Q1987 Q1988 Q1989 Q1990 Q1991 Q1992 Q1993 Q1994 Q1995 Q1996 Q1997 Q1998 Q1999 Q2000

Q2001 Q2002 Q2003 Q2004 Q2005 Q2006 Q2007 Q2008 Q2009 Q2010 Q2011 Q2012 Q2013 Q2014 Q2015 Q2016 Q2017 Q2018 Q2019 Q2020 Q2021 Q2022 Q2023 Q2024 Q2025 Q2026 Q2027 Q2028 Q2029 Q2030 Q2031 Q2032 Q2033 Q2034 Q2035 Q2036 Q2037 Q2038 Q2039 Q2040 Q2041 Q2042 Q2043 Q2044 Q2045 Q2046 Q2047 Q2048 Q2049 Q2050 Q2051 Q2052 Q2053 Q2054 Q2055 Q2056 Q2057 Q2058 Q2059 Q2060 Q2061 Q2062 Q2063 Q2064 Q2065 Q2066 Q2067 Q2068 Q2069 Q2070 Q2071 Q2072 Q2073 Q2074 Q2075 Q2076 Q2077 Q2078 Q2079 Q2080 Q2081 Q2082 Q2083 Q2084 Q2085 Q2086 Q2087 Q2088 Q2089 Q2090 Q2091 Q2092 Q2093 Q2094 Q2095 Q2096 Q2097 Q2098 Q2099 Q2100

Q2101 Q2102 Q2103 Q2104 Q2105 Q2106 Q2107 Q2108 Q2109 Q2110 Q2111 Q2112 Q2113 Q2114 Q2115 Q2116 Q2117 Q2118 Q2119 Q2120 Q2121 Q2122 Q2123 Q2124 Q2125 Q2126 Q2127 Q2128 Q2129 Q2130 Q2131 Q2132 Q2133 Q2134 Q2135 Q2136 Q2137 Q2138 Q2139 Q2140 Q2141 Q2142 Q2143 Q2144 Q2145 Q2146 Q2147 Q2148 Q2149 Q2150 Q2151 Q2152 Q2153 Q2154 Q2155 Q2156 Q2157 Q2158 Q2159 Q2160 Q2161 Q2162 Q2163 Q2164 Q2165 Q2166 Q2167 Q2168 Q2169 Q2170 Q2171 Q2172 Q2173 Q2174 Q2175 Q2176 Q2177 Q2178 Q2179 Q2180 Q2181 Q2182 Q2183 Q2184 Q2185 Q2186 Q2187 Q2188 Q2189 Q2190 Q2191 Q2192 Q2193 Q2194 Q2195 Q2196 Q2197 Q2198 Q2199 Q2200

Q2201 Q2202 Q2203 Q2204 Q2205 Q2206 Q2207 Q2208 Q2209 Q2210 Q2211 Q2212 Q2213 Q2214 Q2215 Q2216 Q2217 Q2218 Q2219 Q2220 Q2221 Q2222 Q2223 Q2224 Q2225 Q2226 Q2227 Q2228 Q2229 Q2230 Q2231 Q2232 Q2233 Q2234 Q2235 Q2236 Q2237 Q2238 Q2239 Q2240 Q2241 Q2242 Q2243 Q2244 Q2245 Q2246 Q2247 Q2248 Q2249 Q2250 Q2251 Q2252 Q2253 Q2254 Q2255 Q2256 Q2257 Q2258 Q2259 Q2260 Q2261 Q2262 Q2263 Q2264 Q2265 Q2266 Q2267 Q2268 Q2269 Q2270 Q2271 Q2272 Q2273 Q2274 Q2275 Q2276 Q2277 Q2278 Q2279 Q2280 Q2281 Q2282 Q2283 Q2284 Q2285 Q2286 Q2287 Q2288 Q2289 Q2290 Q2291 Q2292 Q2293 Q2294 Q2295 Q2296 Q2297 Q2298 Q2299 Q2300

Q2301 Q2302 Q2303 Q2304 Q2305 Q2306 Q2307 Q2308 Q2309 Q2310 Q2311 Q2312 Q2313 Q2314 Q2315 Q2316 Q2317 Q2318 Q2319 Q2320 Q2321 Q2322 Q2323 Q2324 Q2325 Q2326 Q2327 Q2328 Q2329 Q2330 Q2331 Q2332 Q2333 Q2334 Q2335 Q2336 Q2337 Q2338 Q2339 Q2340 Q2341 Q2342 Q2343 Q2344 Q2345 Q2346 Q2347 Q2348 Q2349 Q2350 Q2351 Q2352 Q2353 Q2354 Q2355 Q2356 Q2357 Q2358 Q2359 Q2360 Q2361 Q2362 Q2363 Q2364 Q2365 Q2366 Q2367 Q2368 Q2369 Q2370 Q2371 Q2372 Q2373 Q2374 Q2375 Q2376 Q2377 Q2378 Q2379 Q2380 Q2381 Q2382 Q2383 Q2384 Q2385 Q2386 Q2387 Q2388 Q2389 Q2390 Q2391 Q2392 Q2393 Q2394 Q2395 Q2396 Q2397 Q2398 Q2399 Q2400

Q2401 Q2402 Q2403 Q2404 Q2405 Q2406 Q2407 Q2408 Q2409 Q2410 Q2411 Q2412 Q2413 Q2414 Q2415 Q2416 Q2417 Q2418 Q2419 Q2420 Q2421 Q2422 Q2423 Q2424 Q2425 Q2426 Q2427 Q2428 Q2429 Q2430 Q2431 Q2432 Q2433 Q2434 Q2435 Q2436 Q2437 Q2438 Q2439 Q2440 Q2441 Q2442 Q2443 Q2444 Q2445 Q2446 Q2447 Q2448 Q2449 Q2450 Q2451 Q2452 Q2453 Q2454 Q2455 Q2456 Q2457 Q2458 Q2459 Q2460 Q2461 Q2462 Q2463 Q2464 Q2465 Q2466 Q2467 Q2468 Q2469 Q2470 Q2471 Q2472 Q2473 Q2474 Q2475 Q2476 Q2477 Q2478 Q2479 Q2480 Q2481 Q2482 Q2483 Q2484 Q2485 Q2486 Q2487 Q2488 Q2489 Q2490 Q2491 Q2492 Q2493 Q2494 Q2495 Q2496 Q2497 Q2498 Q2499 Q2500

Q2501 Q2502 Q2503 Q2504 Q2505 Q2506 Q2507 Q2508 Q2509 Q2510 Q2511 Q2512 Q2513 Q2514 Q2515 Q2516 Q2517 Q2518 Q2519 Q2520 Q2521 Q2522 Q2523 Q2524 Q2525 Q2526 Q2527 Q2528 Q2529 Q2530 Q2531 Q2532 Q2533 Q2534 Q2535 Q2536 Q2537 Q2538 Q2539 Q2540 Q2541 Q2542 Q2543 Q2544 Q2545 Q2546 Q2547 Q2548 Q2549 Q2550 Q2551 Q2552 Q2553 Q2554 Q2555 Q2556 Q2557 Q2558 Q2559 Q2560 Q2561 Q2562 Q2563 Q2564 Q2565 Q2566 Q2567 Q2568 Q2569 Q2570 Q2571 Q2572 Q2573 Q2574 Q2575 Q2576 Q2577 Q2578 Q2

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.

productdb is a database in the account1 account. con-product is a container in the productdb database. con-productVendor is a container in the productdb database. con-product is a container in the productdb database.

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

con-productVendor is a container in the productdb database. SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Existing environment.

Litware is a company that is a customer of the productdb database.

- con-productVendor is a container in the productdb database. con-product is a container in the productdb database.

- productdb is a database in the account1 account. HTTP is a protocol that is used to access the productdb database.

- 429 is a status code that is returned by the productdb database. (RU) is a unit of measurement for the productdb database.

is a company that is a customer of the productdb database.

is a company that is a customer of the productdb database.

Litware is a company that is a customer of the productdb database. account2 is a database in the productdb database. IoT is a technology that is used to access the productdb database. IoT is a technology that is used to access the productdb database.

Litware is a company that is a customer of the productdb database.

- account2 is a database in the productdb database.

- account1 is a database in the productdb database.

- con-product is a container in the productdb database.

- Azure Functions is a service that is used to access the productdb database.

- App1 is a container in the productdb database.

- App2 is a container in the productdb database.

- App2 is a container in the productdb database.

is a company that is a customer of the productdb database.

is a company that is a customer of the productdb database.

Litware is a company that is a customer of the productdb database.

IoT is a technology that is used to access the productdb database.

- IoT is a technology that is used to access the productdb database.

- IoT is a technology that is used to access the productdb database.

is a company that is a customer of the productdb database.

is a company that is a customer of the productdb database.

Litware is a company that is a customer of the productdb database.

- IoT is a technology that is used to access the productdb database.

Azure Cosmos DB Core(SQL API) □□□ IoT □□□□ □□□□ telemetry□□ □□□□□□□ □□□□.

□□□□□□□□ 'readings'□ 'devices'□□ □□□ □□□□ □ □□ □□□□.

☐☐ ☐☐☐ ☐☐☐ ☐☐☐☐☐☐

ID

deviceid

□ □ □ □ □

□ □ □

□□□(□□)

-

- ☐

- □□□ID

[illegible]

ID

deviceid

□ □ □

-

- □ □ □ □ □

-

22

11

□□ □ □□□ □□, □□□ □□□□ '□'□ □□□□, □□□ □□□ '□□□'□ □□□□□□.

□□: □□ □□□ 1□□□□.

Answer Area

Statements

To return for all devices owned by a specific email address, multiple queries must be performed

☐

To return deviceid, ownerid, timestamp, and value for a specific metricid, a join must be performed

☐ O



To return deviceid, ownerid, emailaddress, and model, a join must be performed

☐

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: 27

App1 is a .NET application that uses the Azure Cosmos DB for NoSQL API. App1 uses the following schema:

Name	Data type
Col1	Point
Col2	LineString
Col3	Curve
Col4	MultiPolygon
Col5	MultiCurve

App1 uses the following query to retrieve data from the database:
SELECT * FROM c WHERE c.Col1 = 1
A. Col1
B. Col1 OR Col2

C. Col1, Col2, Col3 ☐ ☐ ☐

D. Col1, Col2, Col4 ☐ ☐ ☐

Answer: B (LEAVE A REPLY)

In Azure Cosmos DB for NoSQL, only the Point and LineString data types support spatial indexing. This allows for efficient querying of geospatial data, such as locations (points) and paths (lines). Other data types like Curve, MultiPolygon, and MultiCurve are not supported by spatial indexes in Cosmos DB.

NEW QUESTION: 28

□□ □□ 1 - □□□□ □□□□

Litware, Inc. □ □□□ □□□ □ □□□ □□□□□□□. Litware □ □□□ □□□ □□ □□□□□□ □□ □□□□. □□ □□□ 50□□ □□□□ □□□□ □□□ □□□ □□□ □□ □□□□ □□□□. □ □□□ □□ □□ □□ □□□□ □□□□. □□ □□.

□□□□ □ □□□ □□□ □□. Litware□ □□ □□ □□□ □□□□ □□□□ Azure □□□ □□□□ □□□□.

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.

```
productdb 0 00000 00 000000 0000 0000. con-product 000000 000 00 0000 0000 000000. con-product 0 0 0000 con-product vendor 00 0000 0000. con-product
0 0000 0000 00 0000 0000 00 0000 0000.
```

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

con-productVendor □□□□□ □□□□ □□□□□□ □□□□ □□□ □□□ □□ □□□□□. SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Existing environment.

□ □ □ □ □ □ □ □ .

Litware □ □□□ □□ □□□□ □□□□□□.

- con-productVendor □□□□□ □□ □□□□ □□□□□ con-product □□□□□ □□□ □□□□ □□□□ □□□□.

- 0000 000 0000000 000000 00 HTTP 00 00 429 "000 00 00000" 000 00 000000.

- 429 □□ □□□ □□□□ □ □□□ □□ □□(RU) □□□ □□□ □□□ □□ □□□□□.

☐ ☐ ☐ ☐ .

□ □ □ □ □ □ □

Litware account2 is an Azure Cosmos DB for NoSQL database, which is an iotdb database. iotdb database con-iot1 con-iot2 is an iotdb database.

Litware □ □□ □□ □□ □□□ □□□ □□□□□.

- □□ □□ □□□□ account2□ □□□□□.

- account1□ □□ □□/□□ □□□ □□□□□ □□□□□.

- con-product □□□□□ □□ □□ □□□□ □□□□□.

- Azure Functions ☐ ☐☐☐ ☐☐ ☐☐☐☐ ☐ ☐☐ ☐☐☐☐ ☐ ☐

- □□ □□□□ □□□□ account1□ □□□□ □□□ □ □□ App1□□□ □□ □□□□□□.

```
- App2  00 00 0000 000000 0000 account2 00 0000 000000 0000.
```

```
- App2 account2 000 0 00 DNS 000000 00000 000.
```

□ □ □ □ .

□ □ □ □ □ □ □ □

Litware□ □□□ □□ □□□□ □□ □□□ □□□□□□.

□□□□□ □□ □□ □□□□ □□ □□, □□ □□□ □□□□ □□□□ □□□ □□□ □□□□ □□□□ □□□□□□.

- IoT ☐ ☐ ☐ ☐ ☐ Azure Cosmos DB ☐ ☐ ☐ ☐ ☐ ☐.

- □□□□□ □□□ □□ □□□ □□□□□□.

□ □ □ □ .

□ □ □ □ □ □ □ □

Litware□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □.

- □□ □□□□ □□□□ □□ □□□ □□ □□ □□ □□□ □□□□□.

- □□□□ □□ □□□□ □□□□ □ □□□ □□ □□□ □□□□□□.

- 00 00 00 000 0000 account1 0000 00 App1 000 000 00000000.

- con-product □□□□ □□ □□□ □□□□□ □□□ □ □□ Azure □□□ □□□□□ □□□.

□ □ □ □ □

con-product □ con-productVendor□ □□□□□ □□□□ □□□. □□□□ □□ □□□□ □□ □□□ □□□□ □□ □□□□ □□□.

☐ ☐☐☐☐☐☐ ☐☐ ☐☐☐ ☐☐☐☐☐☐? ☐☐☐☐☐ ☐☐ ☐☐☐☐☐ ☐☐ ☐☐☐☐.

□□: □□ □□□ 1□□□□.

Answer Area



Microsoft

con-product:

▼

Composite on con-productVendor and id

Range on con-productVendor

Spatial on con-productVendor

con-productVendor:

▼

Composite in creditRating and yearFounded

Range on creditRating

Range on yearFounded

Answer:

Answer Area

con-product:

▼

Composite on con-productVendor and id

Range on con-productVendor

Spatial on con-productVendor

con-productVendor:

▼

Composite in creditRating and yearFounded

Range on creditRating

Range on yearFounded

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: 29

□□□ □□

account1□□□ □□□ Azure Cosmos DB for NoSQL □□□ □□□□.

account1 100GB

```
SELECT *
FROM c
WHERE LOWER(c.categoryid) = "hockey"
```

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

	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	☐	☐

Answer:

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	☒	☐

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: 30

Azure Cosmos DB for NoSQL □□□□□ □□□□.

□□□□□ □□ □□ □□□ □□□ □□ □□(Bounded Staleness) □□□ □□□ □□□□□.

□□□□□ □□ □□ □□□ □□□ □ HTTP 429 □□□ □□□□ □□ □□□□□□.

HTTP 429 □□ □□ □□□ □□□ □ □□ □□□ □□□□□? □ □□□ □□□ □□□□ □□□□□.

□□: □□ □□□ 1□□□□.

A. □□ □□ □□□ □□□□ □□□ □□□ □□□□□ □□□□□.

B. □□□□□ □□□ □□□□ □□□□□.

C. □□□□□ □□ □□□□ □□□□□.

D. □□□□□ □□□□□□□ □□ □□□ □□□ □□□□□.

E. □□ □□□□ □□□□□□.

Answer: B,E (LEAVE A REPLY)

[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: 31

□□: □ □□□ □□□ □□□□□ □□□□ □□ □ □ □□□□. □ □□□□ □□□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□□ □ □ □□□ □ □□□, □□□□ □□ □□ □□□ □□□□.

□ □□□□ □□□ □□□ □□□ □□ □□□□ □□□ □ □□□□. □□□ □ □□□□ □□ □□□ □□□□ □□□□.

Azure Cosmos DB for NoSQL □□□□ □□ □□ □□□ □□□□ □□□□□□□□ □□□□.

□□□□□□ □□ □□ □□ □□□ □□□□□ Azure Cosmos DB SDK□ □□□□ □□□□. □□□□ □□ □□□ □□ □□□ □□□□□ □□ □□□.

□□ □□: ConflictResolutionMode□ LastWriterWins□ □□□□ □□□□ □□ □□ □□□ □□□□□.

□□□□ □□□ □□□□□?

- A. □
- B. □□□

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. "

DP-420-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ DP-420-KR □□! DumpTop □ □□ DP-420-KR □□ □□□ □□□□□□, DumpTop DP-420-KR □□ □□□ □□□□□□□□ □□□ □□□ □□□□□. □□□□ □□□ □□□□ □□ DumpTop DP-420-KR □□□ □□□□□. <https://www.dumptop.com/Microsoft/DP-420-KR-dump.html> (281 Q&As Dumps, 30%OFF Special Discount: KrDump)

NEW QUESTION: 32

□□□□ □□

□□□ □□□□□□□ Azure Cosmos DB for NoSQL□ □□□□□□□□ □□□□. □□ □□□□□□□□ □□□ □□ □□□ □□□□ □□□□ □□□□.

Answer Area

Number of containers:

1

2

3

Partition key:

id

name

type

NEW QUESTION: 33

You are configuring an Azure Cosmos DB database. The database is named Account1 and is located in the East US region. The database is configured with the following settings:

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 database is configured with the following partition key:

Partition Key
id
name
type

 The database is configured with the following partition key:

Partition Key
id
name
type

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 database is configured with the following partition key:

Partition Key
id
name
type

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

The database is configured with the following partition key:

Partition Key
id
name
type

NEW QUESTION: 35

□ □ □ □ □

Azure ☐☐☐ ☐☐☐☐☐ ☐☐☐☐.

Azure Cosmos DB for NoSQL □□□ □□ □□□□ □□□□ □□□.

-

- □□□ □□ □□(UDF)

☐ ☐☐☐☐ ☐☐ ☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐.

□ □□□ □□□ □□□□ □□□? □□□□□ □□□□□ □□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

ANSWER AREA

```
"type": "Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/userDefinedFunctions",
```

	"2022-05-15",
"apiVersion":	
"contentVersion":	
"version":	

```
name": "[format('{0}/{1}/{2}/{3}', toLower(parameters('accountName')), parameters('databaseName'), parameters('containerName'), 'udfincometax'))"]
```

```
'properties': {
```

```
"resource": {
```

```
"id": "udfincometax",
```

```
"body": "function tax(income){if(income<1000)return income*0.1;else return income*0.4;}"
```

}

```
[ "[resourceId('Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers', toLower(parameters('accountName'))  
  "dependsOn": [ parameters('databaseName'), parameters('containerName') ] ]"  
  "outputs": {  
    "reference": {  
      type: "string",  
      value: "[resourceId('Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers', toLower(parameters('accountName'))  
        parameters('databaseName'), parameters('containerName'))]"  
    }  
  }  
]
```

Answer:

Microsoft Azure

type": "Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/userDefinedFunctions",

2022-05-15",

apiVersion:

contentVersion:

version:

name": "[format('{0}/{1}/{2}/{3}', toLower(parameters('accountName')), parameters('databaseName'), parameters('containerName'), 'udfincometax'))]"

properties": {

resource": {

id": "udfincometax",

body": "function tax(income){if(income<1000)return income*0.1;else return income*0.4;}"

}

,

dependsOn:

outputs:

reference:

}

]

NEW QUESTION: 36

Azure Cosmos DB Core(SQL) API ☐ ☐ ☐ ☐.

☐ ☐ ☐ ☐ Log Analytics ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐.

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ (RU/s) ☐ ☐ ☐ ☐ ☐ ☐.

☐ ☐ ☐ ☐ ☐ ☐.

Azure ☐

| Category == "ControlPlaneRequests" ☐ ☐

☐ ☐ ☐ ☐ ☐?

A. | OperationName ☐ "AccountUpdateStart" ☐ ☐ ☐ ☐

B. | OperationName ☐ "SqlContainersDelete" ☐ ☐ ☐ ☐

C. | OperationName ☐ "MongoCollectionsThroughputUpdate" ☐ ☐ ☐ ☐

D. | OperationName ☐ "SqlContainersThroughputUpdate" ☐ ☐ ☐ ☐

Answer: D (LEAVE A REPLY)

AzureDiagnostics

| where Category == "ControlPlaneRequests"

| where OperationName startswith "SqlContainersThroughputUpdate"

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/audit-control-plane-logs>

NEW QUESTION: 37

Azure Cosmos DB for NoSQL ☐ ☐ ☐ ☐ ☐ ☐.

accountnumber□□ □□□ □□□□ □□□□□□□ □□□□ □□□□ □□□□ Azure □□□ □□□□ □□□. □ □□□□ SQL □□□ □□□□□□ □□□□□ □□□□. □□ □□□ □□□□ □□□ □□□□ □□□?

A. cmd = "SELECT * FROM Persons p
p.accountnumber = 'accountnumber'□ □□

B. cmd = "SELECT * FROM Persons p
□□□ p.accountnumber□ @accountnumber□ □□□□.

C. cmd = "SELECT * FROM Persons p
p.accountnumber = @accountnumber□ □□

D. cmd = "SELECT * FROM Persons p
□□□ p.□□□□□ "" + □□□□ + ""□□□.

Answer: C ([**LEAVE A REPLY**](#)**)**

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](https://docs.microsoft.com/en-us/azure/cosmos-db/sql/sql-query-parameterized-queries)

NEW QUESTION: 38

□□□ □□

Azure Cosmos DB for NoSQL □□□ □□□ Azure □□□ □□□□.

□□ C# □□□ □□□□ □□□ □□□ □□□□ □□□□□.

Answer Area

Statements

Queries can retrieve items from the Customers container by using the taxId field.

Yes

☒

No

☐

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.

☐☒

NEW QUESTION: 39

Azure Cosmos DB for NoSQL is used to store IoT data. The IoT data is stored in a container named telemetryId. The container has a partition key of telemetryId. The container has a throughput of 1,000 RU/s. The container has a size of 5,000 MB. The IoT data is stored in the container. The IoT data is stored in the container. The IoT data is stored in the container. The IoT data is stored in the container. The IoT data is stored 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"
}
```

The IoT data is stored in the container. The IoT data is stored in the container. The IoT data is stored in the container. The IoT data is stored in the container. The IoT data is stored in the container.

- A. The offerThroughput property is set to 1,000.
- B. The offerThroughput property is set to 5,000.
- C. The deviceId property is set to the partition key.
- D. The deviceId property is set to the container name.

Answer: C (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.

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

NEW QUESTION: 40

□ □ □ □ □

Azure Cosmos DB for NoSQL ☐☐☐ ☐☐☐ ☐ ☐☐ ☐☐☐☐☐

□□□□ □□□□ RU(□□ □□)□ □□□□□□ □□□□ □□□□ □□□.

☐ ☐☐☐ ☐☐ ☐☐☐ ☐☐☐ ☐☐☐? ☐☐☐☐ ☐☐ ☐☐☐☐ ☐☐ ☐☐☐☐.

□□:□□ □□□ 1□□□□.

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	

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

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>

How many Log Analytics workspace can you create in a region?
How many Log Analytics workspace can you create in a region?
How many Log Analytics workspace can you create in a region?

AzureDiagnostics

| "ControlPlaneRequests" |

How many Log Analytics workspace can you create in a region?

A. | OperationName | "SqlContainersCreate" |

B. | OperationName | "SqlContainersThroughputUpdate" |

C. | OperationName | "RegionAddComplete" |

D. | OperationName | "MongoCollectionsThroughputUpdate" |

Answer: B (LEAVE A REPLY)

NEW QUESTION: 42

IoT Hub can connect to Azure Cosmos DB for NoSQL. How many IoT Hub can you create in a region?
How many IoT Hub can you create in a region?
How many IoT Hub can you create in a region?

```
{
  "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"
}
```

How many IoT Hub can you create in a region?
- How many IoT Hub can you create in a region?
- How many IoT Hub can you create in a region?
- How many IoT Hub can you create in a region?
How many IoT Hub can you create in a region?

A. How many IoT Hub can you create in a region?

B. deviceId sensor1Value |

C. deviceId deviceManufacturer |

D. deviceId |

Answer: D (LEAVE A REPLY)

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: Sensor1Value 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: 43

☐☐☐ ☐☐

Azure Cosmos DB for NoSQL ☐☐☐ ☐☐☐ Azure ☐☐☐ ☐☐☐☐.

☐☐ ☐☐☐ ☐☐☐☐☐.

```
QueryDefinition query = new QueryDefinition("SELECT DISTINCT VALUE e.lastname FROM employees e ORDER BY  
e.lastname");  
string continuation = null;
```

```
List<Cars> results = new List<Cars>();  
using (FeedIterator<Cars> resultSetIterator = container.GetItemQueryIterator<Cars>(  
    query, requestOptions: new QueryRequestOptions()  
    {  
        MaxItemCount = -1  
    }  
))  
{  
    while (resultSetIterator.HasMoreResults)  
    {  
        FeedResponse<Cars> response = await resultSetIterator.ReadNextAsync();  
        results.AddRange(response);  
        if (response.Count > 0)  
        {  
            continuation = response.ContinuationToken;  
            break;  
        }  
    }  
}
```



□□ □ □□□ □□, □□□ □□□□ '□'□ □□□□, □□□ □□□ '□□□'□ □□□□□□.
□□: □□ □□□ 1□□□□.

Answer Area

Statements	Yes	No
As written, the query will return results.	☐	☐
The MaxItemCount parameter limits the number of results per query execution.	☐	☐
The ContinuationToken parameter is used to process queries from multiple pages.	☐	☐

Answer:

Answer Area

Statements	Yes	No
As written, the query will return results.	☐	☒
The MaxItemCount parameter limits the number of results per query execution.	☐	☒
The ContinuationToken parameter is used to process queries from multiple pages.	☒	☐

NEW QUESTION: 44

□□□ □□

Azure Cosmos DB Core(SQL API) □□□ □□□□□□□ □□□□.

5,000□□ □□□ □□□ □□ □□ □□□□ □□□ □□□□□□. □ □□□ □□ 25□□ □□□ □□□ □ □□□□□. □ □□ □□□□ emailAddress □□ □□□□□□.

□□ □□ □ □□ □□□ emailAddress □□ □□□□□ □□□.

□□□□ □□ □□ □□□□ □□□□ □□□? □□□□□□ □□ □□□□□ □□□ □□□□□□□□.

□□: □□ □□□ 1□□□□□.

Answer Area

Partition key

companyId

companyId+emailAddress

emailAddress

employeeId

Unique key

companyId

emailAddress

employeeId

Answer:

Answer Area

Partition key

companyId

companyId+emailAddress

emailAddress

employeeId

Unique key

companyId

emailAddress

employeeId

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>

Which statement is true?

```
{
  "orderId" : "d4a91979b-5ead-43a3-b851-add9a71ac4b6",
  "customerId" : "f6e39103-bdc7-4346-9cfb-45daa4b2becf",
  "orderDate" : "2021-09-29",
  "orderItems" : [
    {
      "itemId" : "6c30412f-3cd7-4cab-813c-05942345720d",
      "name" : "blue pen",
      "type" : "pens",
      "count" : 10,
    },
    ...
  ],
  "total" : 12345,
  "status" : "ordered"
}
```

Which statement is true?
A. The order is for a customer with ID f6e39103-bdc7-4346-9cfb-45daa4b2becf.
B. The order is for a customer with ID d4a91979b-5ead-43a3-b851-add9a71ac4b6.
C. The order is for a customer with ID 6c30412f-3cd7-4cab-813c-05942345720d.
D. The order is for a customer with ID 12345.

- A. SQL queries can be used to query data from the database.
- B. The order is for a customer with ID f6e39103-bdc7-4346-9cfb-45daa4b2becf.
- C. The order is for a customer with ID d4a91979b-5ead-43a3-b851-add9a71ac4b6.
- D. The order is for a customer with ID 6c30412f-3cd7-4cab-813c-05942345720d.

Answer: (SHOW ANSWER)

After processing items in the change feed, you can build a materialized view and persist aggregated values back in Azure Cosmos DB. If you're using Azure Cosmos DB to build a game, you can, for example, use change feed to implement real-time leaderboards based on scores from completed games.

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/change-feed-design-patterns#high-availability>

NEW QUESTION: 46

Which statement is true?
A. The order is for a customer with ID f6e39103-bdc7-4346-9cfb-45daa4b2becf.
B. The order is for a customer with ID d4a91979b-5ead-43a3-b851-add9a71ac4b6.
C. The order is for a customer with ID 6c30412f-3cd7-4cab-813c-05942345720d.
D. The order is for a customer with ID 12345.

```
"userid": "user1@contoso.com",
"id": "e379aea5-63f5-4623-92e9b-4cd9b33b91d5",
"orderdate": "2022-11-01T02:00:00",
"items": [
  {
    "sku": "11111",
    "qty": 1
  },
  {
    "sku": "11112",
    "qty": 5
  }
]
]
```



SKU 11111 is the only SKU with a quantity of 1. The order date is 2022-11-01T02:00:00. The user ID is e379aea5-63f5-4623-92e9b-4cd9b33b91d5.

Answer Area

```
SELECT c.userid
FROM c
WHERE EXISTS(
  SELECT VALUE n
  FROM
  WHERE
```



	▼
c.items	
c.items.sku	
n IN c.items	

	▼
n	
n.items.sku	
n.sku	

= "11111"

Answer:

Answer Area

```
SELECT c.userid
FROM c
WHERE EXISTS (
    SELECT VALUE n
    FROM
        

c.items



c.items.sku



n IN c.items


    WHERE
        

n



n.items.sku



n.sku


        = "11111"
)
```



DP-420-KR 100% 100% 100% 100% DumpTop 100% 100% 100% DP-420-KR 100%! DumpTop 100% 100% DP-420-KR 100% 100% 100% 100%, DumpTop DP-420-KR 100% 100% 100% 100% 100% 100% 100%. 100% 100% 100% 100% 100% DumpTop DP-420-KR 100% 100% 100%. <https://www.dumptop.com/Microsoft/DP-420-KR-dump.html> (281 Q&As Dumps, 30%OFF Special Discount: KrDump)

NEW QUESTION: 47

Azure Cosmos DB 100% 100% 100% 100%.

100% 100% 100% 100% 100% 100% 100% 100% 100% container1 100% 100% 100% 100% 100%.

100%100% 100% 100% 100% 100%. 100% 100% 100% 100% 100%.

- 100% 100% 100%.

- 100% 100% 100%.

100% 100% 100% 100% 100%? 100% 100% 100% 100%.

100: 100% 100% 100%.

A. 100%

B. 100% 100%

C. 100% 100%

D. 100%

Answer: B,D (LEAVE A REPLY)

High cardinality: Choosing a partition key with high cardinality means having a large number of unique values distributed across the dataset. This helps in distributing the data evenly across multiple partitions, preventing hotspots and ensuring efficient scalability.

Static: Selecting a static partition key means choosing a key that does not frequently change.

This minimizes the need for updates to the partition key value and reduces maintenance effort. It also helps in maintaining a consistent distribution of data across partitions over time.

NEW QUESTION: 48

Scenario: You are building a data pipeline that processes data from Azure Cosmos DB Core (SQL) API container1 and outputs the data to Azure Stream Analytics container1. The data is stored in Azure Blob Storage and is processed by Azure Synapse. The data is then loaded into a data warehouse.

What should you do to ensure that the data is loaded into the data warehouse in the correct order?

A. Use a partition key that is unique for each record.

B. Use a static partition key.

Answer: B (LEAVE A REPLY)

Instead create an Azure function that uses Azure Cosmos DB Core (SQL) API change feed as a trigger and Azure event hub as the output.

The Azure Cosmos DB change feed is a mechanism to get a continuous and incremental feed of records from an Azure Cosmos container as those records are being created or modified.

Change feed support works by listening to container for any changes. It then outputs the sorted list of documents that were changed in the order in which they were modified. The following diagram represents the data flow and components involved in the solution:



Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/changefeed-ecommerce- solution>

NEW QUESTION: 49

App1 is an application that uses Azure Cosmos DB for NoSQL. App1 is configured to use the integrated cache. The application is running on a server that is connected to the Azure Cosmos DB account. The application is using the integrated cache to store data. The application is using the integrated cache to store data. The application is using the integrated cache to store data.

What should you do to ensure that the application is using the integrated cache?

A. Azure Cosmos DB integrated cache is enabled.

B. App1 is configured to use the integrated cache.

C. App1 CosmosClient is configured to use the integrated cache.

D. Azure Cosmos DB integrated cache is disabled.

Answer: C (LEAVE A REPLY)

Because the integrated cache is specific to your Azure Cosmos DB account and requires significant CPU and memory, it requires a dedicated gateway node. Connect to Azure Cosmos DB using gateway mode.

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

NEW QUESTION: 50

Azure Cosmos DB for NoSQL Container1 100 (RU/s)

□□□□1□ □□ □□□□ □□□□ □□□□.

Container1 □ □□ □ □□ □□ □□□ □□□ □□□□□?

- A.** 300 RU/s
B. 1,000 RU/s
C. 30,000 RU/s
D. 100,000 RU/s

Answer: ([SHOW ANSWER](#))

Azure Cosmos DB containers can accumulate burst capacity when they are under-utilized.

Containers that have provisioned throughput and are using a single physical partition can accumulate up to 5 minutes' worth of their provisioned throughput in burst capacity, up to a maximum of 300 RU/s per physical partition.

Given that Container1 has a provisioned throughput of 100 RU/s and uses a single physical partition, it can accumulate up to 300 RU/s in burst capacity (3 times the provisioned throughput).

This burst capacity can be used during short periods of increased load when the container needs more throughput than is provisioned.

NEW QUESTION: 51

□ □ □ □ □ □ □ □

Azure ☐☐☐ Azure Cosmos DB for NoSQL ☐☐☐ ☐☐☐☐ ☐☐☐☐ DB1☐☐☐☐☐☐☐☐ KV1☐☐☐☐ Azure Key Vault☐☐☐☐.

DB1 Container1 Always Encrypted

□□ □ □□ □□□ □□□□ □□□□ □□□? □□□ □□□□, □□ □□□□ □□□ □□□ □□ □□□ □□□□ □□□□ □□□□ □□□□.

Actions	Answer Area
Create a customer-managed key.	1
Create Container1.	2
Store the customer-managed key in KV1.	3
Store the database encryption key in KV1.	4
Create a data encryption key.	

Answer:

Actions

Create a customer-managed key.

Create Container1.

Store the customer-managed key in KV1.

Store the database encryption key in KV1.

Create a data encryption key.

Answer Area

1 Create a customer-managed key.

2 Store the customer-managed key in KV1.

3 Create a data encryption key.

4 Create Container1.

NEW QUESTION: 52

account1 is an Azure Cosmos DB for NoSQL database, and App1 is an Azure application that uses the Azure Cosmos DB .NET SDK to connect to account1. App1 is configured to use the Azure Cosmos DB .NET SDK to connect to account1. App1 is configured to use the Azure Cosmos DB .NET SDK to connect to account1.

Azure Cosmos DB .NET SDK is configured to connect to App1.

App1 is configured to connect to App1. App1 is configured to connect to App1. App1 is configured to connect to App1.

- App1 is configured to connect to App1. App1 is configured to connect to App1.

App1 is configured to connect to App1.

- App1 is configured to connect to App1.

App1 is configured to connect to App1?

A. SDK is configured to connect to App1.

B. SDK is configured to connect to App1. App1 is configured to connect to App1.

C. SDK is configured to connect to App1, App1 is configured to connect to App1, App1 is configured to connect to App1.

D. SDK is configured to connect to App1, App1 is configured to connect to App1, App1 is configured to connect to App1.

Answer: D ([LEAVE A REPLY](#))

To satisfy all constraints, you should enable multi-region writes on the Azure Cosmos DB account

[assume this already done] and configure the ApplicationRegion property in the .NET SDK client initialization.

Resuming Writes Instantly: With multi-region writes enabled, every region becomes a valid write location. If the primary region experiences an outage, active-active capability prevents the database from performing a traditional database-level failover. This lowers write recovery time to zero since alternate regions can immediately ingest traffic.

*-> Optimized Read and Write Latency: Passing the ApplicationRegion property (e.g., Regions.WestEurope) to the SDK allows it to locate the nearest active geographic replica. This guarantees low-latency local reads and writes within West Europe and Southeast Asia.

*-> Minimal Code Changes: Instead of hardcoding complex prioritized lists via ApplicationPreferredRegions, setting the single ApplicationRegion string automatically creates an ordered fallback list behind the scenes based on geographic proximity.

Reference:

<https://learn.microsoft.com/en-us/azure/cosmos-db/troubleshoot-dotnet-sdk-slow-request>

NEW QUESTION: 53

Account1(SMB) is an Azure Cosmos DB for NoSQL database, and App1 is an Azure application that uses the Azure Cosmos DB .NET SDK to connect to Account1.

App1 is configured to connect to App1. App1 is configured to connect to App1.

- 100,000 items per container. 100,000 items per container.
- 1,000 items per container.
- 2KB items per container.
- 100,000 items per container.
- 100,000 items per container.
- 100,000 items per container.
- 100,000 items per container.
- 100,000 items per container.
- 100,000 items per container.
- 100,000 items per container.
- 100,000 items per container.
- 100,000 items per container.
- 100,000 items per container.

100,000 items per container.

- A. 100,000 items per container. 100,000 items per container.
- B. 100,000 items per container. 100,000 items per container.
- C. 100,000 items per container. 100,000 items per container.
- D. 100,000 items per container. 100,000 items per container.

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: 54

100,000 items per container.

account1 100,000 items per container.

account1 100,000 items per container.

Azure CLI (CLI) 100,000 items per container.

100,000 items per container.

Answer Area

az backup container
az cosmosdb restore
az cosmosdb update

```
--resource-group "RG1" \
--name "account1" \
```

--backup-policy-type "Continuous" --continuous-tier "Continuous7Days"
--backup-policy-type "Periodic"
--backup-retention 7 --backup-interval 604800

Answer:

Answer Area

az backup container

az cosmosdb restore

az cosmosdb update

--resource-group "RG1" \

--name "account1" \

--backup-policy-type "Continuous" --continuous-tier "Continuous7Days"
--backup-policy-type "Periodic"
--backup-retention 7 --backup-interval 604800

NEW QUESTION: 55

Azure Cosmos DB for NoSQL ☐ ☐ Azure ☐ ☐ ☐ ☐.

☐ ☐ ☐ ☐ SQL ☐ ☐ ☐ ☐. ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐.

- ☐ ☐ ☐ ☐ ☐ UUC(☐ ☐ ☐ ☐)☐ ☐ ☐ ☐.

☐ (UTC)☐ ISO 8601 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐.

- ☐ ☐ ☐ ☐ ☐.

☐ ☐ ☐ ☐ ☐?

A. ☐ ☐ ☐ ☐

B. ☐ ☐ ☐ ☐ ☐ ☐()

C. ☐ ☐ ☐ ☐ ☐ ☐()

D. DateTimePart()

Answer: B (LEAVE A REPLY)

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: 56

account1☐ ☐ Azure Cosmos DB ☐ ☐ ☐ ☐.

☐ ☐ ☐ ☐ account1☐ ☐ ☐ ☐ ☐ ☐ VPN☐ ☐ ☐ ☐.

☐ ☐ ☐ ☐ ☐?

A. ☐ ☐ ☐ ☐ ☐

B. ☐ ☐ ☐ ☐ ☐

C. Azure Synapse ☐ ☐

Answer: A (LEAVE A REPLY)


```
{
  "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
    }
  ]
}
```

□□ □□ □□□ 100□□ □ □□□ □□ □□, □□(GB) □ □□ □□ □□□□ □□□.
□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.
□□: □□ □□□ 1□□□□.

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
```

Answer:

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
```



NEW QUESTION: 59

Azure Cosmos DB for NoSQL `Container1` has a `DefaultTimeToLive` property set to `-1`.

`Container1` has a `ttl` property set to `1000`.

What is the result of the `ttl` property?

A. The item will be deleted after 1,000 seconds (approximately 16.7 minutes) from the time it was last modified.

B. The item will be deleted after 1,000 minutes (approximately 16.7 hours) from the time it was last modified.

C. The item will be deleted after 1,000 hours (approximately 41.7 days) from the time it was last modified.

D. The item will be deleted after 1,000 days (approximately 2.7 years) from the time it was last modified.

Answer: C (LEAVE A REPLY)

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: 60

Azure Cosmos DB for NoSQL supports ACID (atomicity, consistency, isolation, durability) transactions.

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>

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>

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>

Answer: A (LEAVE A REPLY)

Container1 Azure Cosmos DB for NoSQL Container1

```
{
  "id": "Family1",
  "parents": [
    { "familyName": "Smith", "givenName": "Robert" },
    { "familyName": "Jones", "givenName": "Sarah" }
  ],
  "children": [
    {
      "familyName": "Smith",
      "givenName": "Jesse",
      "gender": "female", "grade": 1,
      "pets": [
        { "givenName": "Mo" },
        { "givenName": "Nora" }
      ]
    },
    {
      "familyName": "Jones",
      "givenName": "Lisa",
      "gender": "female",
      "grade": 8
    }
  ],
  "address": { "state": "WA", "county": "King", "city": "Seattle" },
  "creationDate": 1431620462,
  "isRegistered": false
}
```

Family1 Given Family
?
1

Answer Area

```
SELECT c.givenName, c.familyName
FROM Families f
JOIN c IN f.
```

▼

children
familyName
id
parents

WHERE

▼

c.id
f.id
familyName
name

= 'Family1'



Answer:

Answer Area

```
SELECT c.givenName, c.familyName
```

```
FROM Families f
```

```
JOIN c IN f.
```

	▼
children	
familyName	
id	
parents	

```
WHERE
```

	▼
c.id	
f.id	
familyName	
name	

```
= 'Family1'
```


Reference:
<https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-setup-cmk>

NEW QUESTION: 63

Which of the following queries will benefit from a composite index?

```
SELECT * FROM c
WHERE c.sensor = "TEMP1"
AND c.value < 22
AND c.timestamp >= 1619146031231
```

Which of the following queries will benefit from a composite index?

Which of the following queries will benefit from a composite index?

- A. (ASC, ASC) or (ASC, ASC) or (ASC, ASC) or (ASC, ASC)
- B. (ASC, ASC, ASC) or (DESC, DESC, DESC) or (ASC, DESC, DESC) or (ASC, DESC, DESC)
- C. (ASC, ASC) or (ASC, ASC) or (ASC, ASC) or (ASC, ASC)
- D. (ASC, ASC, ASC) or (ASC, ASC)

Answer: A (LEAVE A REPLY)

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: 64

Which of the following queries will benefit from a composite index?

Which of the following queries will benefit from a composite index?

Which of the following queries will benefit from a composite index?

Which of the following queries will benefit from a composite index?

Which of the following queries will benefit from a composite index?

Which of the following queries will benefit from a composite index?

A.

B.

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: 65

□□□ □ □□ □□

Azure Cosmos DB for NoSQL □□□□□□□ □□□, □ □□□□□□□□ Container1□□□ □□□ □□□□□ □□□□. Container1□□ □□□ □□ □□□ □□ □□ □□□ □□□□□.

```
{
  "id": "1",
  "address": {
    "city": "London"
  },
  "age": 45
}
{
  "id": "2",
  "address": {
    "city": "Paris"
  },
  "age": 36
}
```

40□ □□ □□ □□□ □□ □□□ □□□ □□□□ □□□. □□□ □□□□ □□ □□□□ □□□□ □□ □ □□ □□□ □□□□□? □□□ □□□□□ □□ □□□□ □□□ □□□□ □□□□ □□□ □□□ □□□□□ □□□□□□.

Statements

WHERE "40" < c.age

FROM Container1

SELECT c.address.city, c.age

SELECT c.address.city

SELECT VALUE {c.address.city, c.age}

FROM c

WHERE c.age > 40

Answer Area

1

2

3

Answer:

Statements

WHERE "40" < c.age

FROM Container1

SELECT c.address.city, c.age

SELECT c.address.city

SELECT VALUE {c.address.city, c.age}

FROM c

WHERE c.age > 40

Answer Area

1SELECT c.address.city, c.age

2FROM c

3WHERE c.age > 40

NEW QUESTION: 66

□□□ □□

□□ □□□ □□ □□ Azure Cosmos DB □□(account1)□ □□□ □□ □□□ □□□ □□(strong)□□ □□□□ □□□□.

App1□□□□ □□ □□□, □ □□ □□ □□□ □□□ □□□□□ □□□□ □□□□.

App1□ □□ □ □□ □□□ □□□ □□□□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.

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 will be performed by the replica:

To which App1 is connected

That has the lowest estimated latency to the client

That responds first

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 will 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.

NEW QUESTION: 67

Azure **account1** is an Azure Cosmos DB for NoSQL database with strong consistency and multiple regions. **Container1** is a container in **account1** with 1KB of data. **Container1** is replicated to three regions: East US, Central US, and West US. The round-trip time (RTT) between East US and Central US is 96 ms, between East US and West US is 136 ms, and between Central US and West US is 54 ms. What is the maximum write latency for **Container1**?

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

account1 is an Azure Cosmos DB for NoSQL database with strong consistency and multiple regions. **Container1** is a container in **account1** with 1KB of data. **Container1** is replicated to three regions: East US, Central US, and West US. The round-trip time (RTT) between East US and Central US is 96 ms, between East US and West US is 136 ms, and between Central US and West US is 54 ms. What is the maximum write latency for **Container1**?

- A. 150ms
- B. 232ms
- C. 282ms
- D. 572ms

Answer: C (LEAVE A REPLY)

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 + constant 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: 68

Container1

Azure **Container1** is an Azure Cosmos DB for NoSQL database with strong consistency and multiple regions. **Container1** is replicated to three regions: East US, Central US, and West US. The round-trip time (RTT) between East US and Central US is 96 ms, between East US and West US is 136 ms, and between Central US and West US is 54 ms. What is the maximum write latency for **Container1**?

Container1 is a container in **Container1** with 1KB of data. **Container1** is replicated to three regions: East US, Central US, and West US. The round-trip time (RTT) between East US and Central US is 96 ms, between East US and West US is 136 ms, and between Central US and West US is 54 ms. What is the maximum write latency for **Container1**?

Container1 is a container in **Container1** with 1KB of data. **Container1** is replicated to three regions: East US, Central US, and West US. The round-trip time (RTT) between East US and Central US is 96 ms, between East US and West US is 136 ms, and between Central US and West US is 54 ms. What is the maximum write latency for **Container1**?

Answer: C (LEAVE A REPLY)

Answer Area

...

partitionKey: {
 paths: [
 '/TenantId',
 '/UserId'
]
 kind:

Hash
MultiHash
Range

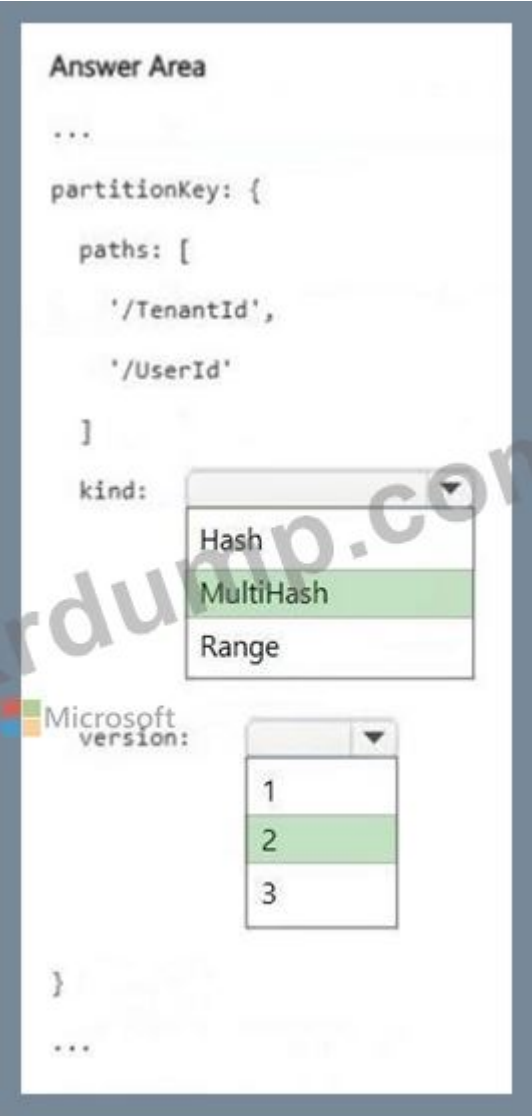
 Microsoft version:

1
2
3

}

...

Answer:



NEW QUESTION: 69

account1 is an Azure Cosmos DB for NoSQL container. You are using the Azure Cosmos DB .NET SDK to create a new container named Container1. Which property should you set to ensure that Container1 has a default time to live of 1 hour?

- A. DefaultTimeToLive -1
- B. DefaultTimeToLive null
- C. TimeToLivePropertyPath null
- D. TimeToLivePropertyPath 0

Answer: B (LEAVE A REPLY)

NEW QUESTION: 70

Spark SQL has a function called explode. Which of the following is the correct syntax for using the explode function?

- A. explode()
- B. posexplode()
- C. explode_col()
- D. explode_col()

Answer: (SHOW ANSWER)

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

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

NEW QUESTION: 71

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □ □□□□ □□□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□□ □ □ □□□ □ □□□, □□□□ □□ □□ □□□ □□□□.

□ □□□□ □□□ □□□ □□□ □□ □□□□ □□□ □ □□□□. □□□□ □ □□□□ □□ □□□ □□□□ □□□□.

Azure Cosmos DB for NoSQL □□□ container1□□□ □□□ □□□□□ □□□□.

Azure Stream Analytics □□□□ □□ □□□□ □□□ □ □□□ container1□ □□□ □□□□ □□□.

□□ □□: Azure Cosmos DB for NoSQL □□ □□□ □□□□, Azure □□□ □□□ □□□□ □□□□ Azure Synapse □□□□□□ □□□□□.

□□□ □□□ □□□□□?

A. □□□

B. □

Answer: (SHOW ANSWER)

NEW QUESTION: 72

□□□ customers□□ □□□ □□□□□ □□□ Azure Cosmos DB for NoSQL □□□ □□□□ □□□□.

App1□□□ □□□□□□□□ □□ □□□ □□□□ □□□, □□ □□□ □□□ □ □□□ □□□ □□□.

□□ □□□ □□□□□. TTL(Time to Live) □□□□ □□□□ □□□□□.

□□□ □□ □□□ □□ □□□.

* □□ □ □□□□ □□□ □□ □□□ 10□ □□□ □□□□□.

* □□ □□□ □□ □□□ □□□□ □□□□.

□1□ □□ □□□ □□ □□□ □□□ □□□ □□ □ □□ □□□ □□□□ □□□?

□□□ □□□ □□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

A. □□□ □□□□ □□□□□.

B. DELETE □□ UPDATE□ □□□□□.

C. □□□ □□□ □□□□ □□□□□.

D. TTL□ □□□□□□

E. □□ □□□□ □□□□□.

Answer: (SHOW ANSWER)

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:

NEW QUESTION: 73

QUESTION 73

App1, App2, App3 are all applications that use Azure Cosmos DB for NoSQL. They all use the same database. Which consistency level should be used?

- App1: The application is a web application that requires high availability and low latency. It is a read-heavy application that does not require strong consistency.
- App2: The application is a web application that requires high availability and low latency. It is a read-heavy application that does not require strong consistency.
- App3: The application is a web application that requires high availability and low latency. It is a read-heavy application that does not require strong consistency.

Which consistency level should be used? The application is a web application that requires high availability and low latency. It is a read-heavy application that does not require strong consistency.

Which consistency level should be used? The application is a web application that requires high availability and low latency. It is a read-heavy application that does not require strong consistency. The application is a web application that requires high availability and low latency. It is a read-heavy application that does not require strong consistency. The application is a web application that requires high availability and low latency. It is a read-heavy application that does not require strong consistency.

Consistency levels	Answer Area
Bounded Staleness	
Consistent Prefix	App1:
Eventual	App2:
Session	App3:
Strong	

Answer:

Consistency levels	Answer Area
Bounded Staleness	
Consistent Prefix	App1: Session
	App2: Strong
	App3: Eventual

NEW QUESTION: 74

QUESTION 74

□□

Litware, Inc. □ □□□ □□□ □ □□□ □□□□□□□. Litware □ □□□□ □□□ □□ □□□□□□ □□ □□□□. □□ □□□ 50□□ □□□□ □□□□ □□□ □□□ □□ □□□□ □□□□. □ □□□ □□ □□ □□ □□□□ □□□□. □□ □□. □□□□ □ □□□ □□□ □□. Litware □ □□ □□ □□□ □□□□ □□□□ Azure □□□ □□□□ □□□□.

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.

productdb □ □ □□□□□ □□ □□□□□ □□□□ □□□□. con-product □□□□□□ □□□ □□ □□□□ □□□□ □□□□□. con-product □ □ □□□□ con-productvendor □□ □□□□ □□□□. con-product □ □□□□ □□□□ □□ □□□□ □□□ □□ □□□ □□□□.

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

con-productVendor □□□□□ □□□□ □□□□ □□ □□□□ □□□ □□□ □□ □□□□□. SELECT * FROM con-productVendor pv ORDER BY pv.creditRating, pv.yearFounded Existing environment. □□ □□□ □□□.

Litware □ □□□ □□ □□□□ □□□□□□.

- con-productVendor □□□□□ □□ □□□□ □□□□□ con-product □□□□□ □□□ □□□□ □□□□ □□□□.
 - □□□□ □□□ □□□□□□ □□□□□ □□ HTTP □□ □□ 429 "□□□□ □□ □□□□" □□□ □□ □□□□□.
 - 429 □□ □□□ □□□□ □ □□□ □□ □□(RU) □□□ □□□ □□□ □□ □□□□□.
- □□.

□□□ □□ □□

Litware □ account2□□ □□□ □□□ Azure Cosmos DB for NoSQL □□□ □□□ □□□□, □ □□□□ iotdb□□ □□□ □□□□□□□□ □□□ □□□□. iotdb □□□□□□□□□ con-iot1□ con-iot2□□ □□□ □□□ □ □ □□ □□□ □□□□□.

Litware □ □□□ □□ □□ □□□ □□□ □□□□□.

- □□ □□ □□□□ account2□ □□□□□.
- account1□ □□ □□/□□ □□□ □□□□□ □□□□□.
- con-product □□□□□ □□ □□ □□□□ □□□□□.
- Azure Functions □ □□□□ □□ □□□□□ □□ □□□ □□□ □□□□□ □□ □ □□□□.
- □□ □□□□ □□□□ account1□ □□□□ □□□ □ □□ App1□□□□ □□ □□□□□.
- App2□□ □□ □□□□ □□□□□ □□□□ account2□ □□□□ □□□□□ □□□.
- App2□ account2□ □□□ □ □□ DNS □□□□□□ □□□□□ □□□.

□□ □□.

□□□□ □□□□

Litware □ □□□ □□ □□□□ □□ □□□ □□□□□□.

□□□□□ □□ □□ □□□□ □□ □□, □□ □□□ □□□□ □□□□ □□□ □□□ □□□□ □□□□□□.

- IoT □□ □□□ □□ Azure Cosmos DB □□□ □□ □□□□ □□□□□.
- □□□□□ □□□ □□ □□□ □□□□□□.

□ □ □ □

□ □ □ □ □ □ □ □ □ □

Litware□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ .

- 00 0000 0000 00 000 00 00 00 000 00000.
- 0000 00 0000 0000 0 00 00 000 000000.
- 00 00 00 000 0000 account10 0000 00 App1 000 000 00000000.
- con-product 0000 00 000 00000 0000 0 00 Azure 000 000000 000.

□ □ □ □ □

con-iot1□ con-iot2□ □□□ □□□□□.

□ □□□□□ □□ □□ TTL(Time to Live) □□□ □□□□ □□□. □□□□ IoT □□ □□ □□ □□□ □□□□ □□□.

□□ □□□ □□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

Answer Area  Microsoft

con-iot1:

	▼
Off	
On (no default)	
On (1 second)	
On (3,600 seconds)	

con-iot2:

	▼
Off	
On (no default)	
On (1 second)	
On (3,600 seconds)	

Answer:

Answer Area

con-iot1:

▼

Off

On (no default)

On (1 second)

On (3,600 seconds)

con-iot2:

▼

Off

On (no default)

On (1 second)

On (3,600 seconds)

Explanation:

Box 1: On (3,600 seconds)

Note: Litware identifies the following IoT telemetry requirements:

Automatically delete items in con-iot1 after one hour unless a per-item time to live is set.

Box 2: Off

Ensure that the items in con-iot2 persist regardless of the per-item time to live setting.

Reference:

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

NEW QUESTION: 75

_____ Azure _____ account1_____ _____ Azure Cosmos DB for NoSQL _____ _____ _____.

account1_____ _____, D4s SKU_____ _____ _____ _____.

account1_____ _____ _____ _____ _____ _____ _____ _____ _____ _____.

- _____ _____ _____ _____, _____ _____ _____ _____ _____.

_____ _____.

- _____ _____ _____ _____, _____ _____ _____ _____.

_____ _____.

☐ ☐☐☐☐☐ ☐☐☐☐☐ ☐☐☐ ☐☐ ☐☐☐? ☐☐☐☐☐ ☐☐☐☐☐ ☐☐☐ ☐☐☐☐☐.

□□: □□ □□□ 1□□□□.

Answer Area

To maximize how long returned items **remain in the**
integrated cache:

Decrease the value of MaxIntegratedCacheStaleness in the point read request.
Increase the SKU of the gateway.
Increase the value of MaxIntegratedCacheStaleness in the point read request.

To maximize the cost benefits of the integrated cache, set the consistency level to:

Consistent prefix
Session
Strong

Answer:

Answer Area

To maximize how long returned items remain in the integrated cache:

Decrease the value of MaxIntegratedCacheStaleness in the point read request.
Increase the SKU of the gateway.
Increase the value of MaxIntegratedCacheStaleness in the point read request.

To maximize the cost benefits of the integrated cache, set the consistency level to:

- Consistent prefix
- Session
- Strong

NEW QUESTION: 76

□ □ □ □ □

Azure Cosmos DB for NoSQL □□□ □□□□.

```
container1□□□ □□□ □□□□□□ □□□ □□□□□□. container1 □□□□□□□ name□ age□□ □ □□ □□□ □□ □□□□ □□□□□□.
```

□□ □□□□ □□□□ □□□ □□□□¹□□ □□ □□□ □□ □□□□□. □□□ □□□ □□□□.

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

□□□□1□ □□ □□□ □□□ □□□ □□□□ □□□. □□□□ □□ □□ □□□ □□□□ □□□.

- □□□□ □□□□ □□ □□ □□ □□□□□□.

- etag □□□ □□□□□ □□□□□ □□□□.

□□□ □□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

Answer Area

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

Answer:

Answer Area

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



DP-420-KR ☐☐ ☐☐☐ ☐☐☐☐☐ ☐☐ DumpTop ☐☐ ☐☐☐☐ ☐☐☐ DP-420-KR ☐☐! DumpTop ☐ ☐☐ **DP-420-KR** ☐☐ ☐☐☐ ☐☐☐☐☐☐, DumpTop DP-420-KR ☐☐ ☐☐☐ ☐☐☐☐☐☐☐☐☐ ☐☐☐ ☐☐☐☐☐☐☐. ☐☐☐☐☐ ☐☐☐☐☐☐☐ ☐☐ DumpTop DP-420-KR ☐☐☐☐☐☐☐☐. <https://www.dumptop.com/Microsoft/DP-420-KR-dump.html> (281 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 77

Azure Cosmos DB for NoSQL ☐☐☐ ☐☐☐☐ ☐☐☐☐

□□□ □□□ □ □□□□□□ □□□ □□□ □□□□, □□□ □□□□ □□□ □□□□ □□□□ □□□.

□□□□ □□□ □□□□ □□□ □ □□ □□□□ □□□ □□□□□? □ □□□ □□□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

A. ☐☐ ☐☐☐ ☐☐ ☐☐ ☐☐☐☐ ☐☐☐☐

B. ☐☐☐ ☐☐

C. ☐☐ ☐☐☐☐☐ 25TB ☐ ☐☐☐ ☐☐

D. ☐☐ ☐☐☐ ☐☐ ☐☐ ☐☐☐ ☐☐☐☐

E. ☐☐☐ ☐☐ ☐☐ ☐☐

Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 78

□□ □□ □□□ □□□□□ □□□□□ Azure Cosmos DB for NoSQL □□□□□□□ □□□□.

Name	Partition key
Products	ProductKey
OrderItems	ProductKey, OrderItemKey

□□ □□□□□□ □□ □□□ □□□ □□, □□ □□□□ □□ □□ □□□ □□ □□□ □□□□□□ □□□□ □□□.

□□□ □□□□ □□□?

A. ☐☐ ☐☐

B. ☐ ☐ ☐

C. ☐ ☐ ☐ ☐ ☐

D. ☐☐ ☐☐☐☐

Answer: A (LEAVE A REPLY)

To ensure the product name stays synchronized between your `Products` and `OrderItems` containers in Azure Cosmos DB, you should implement a Change Feed-driven microservice.

Because Cosmos DB is a non-relational database, it does not support traditional SQL "joins" or automatic cascading updates across containers. You must handle this "denormalization" logic yourself.

The Core Constraint

OrderItems has a composite or different partition key logic (ProductKey, OrderItemKey).

Even if they share ProductKey, a single update in Products cannot automatically lock or update rows in OrderItems across different logical partitions.

Required Components

1. Enable Change Feed on the "Products" Container

This acts as a persistent log of all changes (inserts and updates).

It triggers your logic whenever a `ProductName` is modified.

2. Azure Functions (with Cosmos DB Trigger)

The easiest way to process the Change Feed.

Trigger: Monitors the Products container.

Reference:

<https://devblogs.microsoft.com/cosmosdb/data-modeling-and-partitioning-for-relational-workloads>

NEW QUESTION: 79

□ □ □ □ □ □ □ □

□□□ Azure □□□□ □□ □□ □□□ □□□□ □□□□ □□□□.

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

CosmosDB1□□ □□□ □□□□ □□□ Container1□□□ □□□ □□□□□ □□□ □□□□□.

Container1 ☐ Azure Cosmos DB ☐ ☐ Always Encrypted ☐ ☐☐☐☐ ☐☐☐ ☐

□□ □ □□ □□□ □□□□ □□□□ □□□? □□□ □□□□□, □□ □□□□ □□□ □□□ □□ □□□ □□ □□□ □□□□□ □□□□□.

Actions	Answer Area
Create a policy in Container1.	1
Generate a database encryption key (DEK).	2
Generate a customer-managed key in KV1.	3
Store the database encryption key (DEK) in KV1.	
Create Container1.	

Answer:

Actions	Answer Area
Create a policy in Container1.	1 Generate a database encryption key (DEK).
Generate a database encryption key (DEK).	2 Create Container1.
Generate a customer-managed key in KV1.	3 Create a policy in Container1.
Store the database encryption key (DEK) in KV1.	
Create Container1.	

NEW QUESTION: 80

account1□□□ □□□ Azure Cosmos DB □□□ □□□□.

□□□ □□ □□ □□□□ □□¹□ □□□□ □□ □□ □□□□.

□□ □□ □□□ □□□ □□□□ □□ □□□□□ □□□□□.

account1 ☐ Azure AD ID ☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐.

□□ □□ □□□ □□□□ □□□?

A. □ □ □ □ □ □ □ □ □ □

B. ☐☐ ☐☐ ☐☐☐☐

C. □□□ □□ ID

D. ☐☐☐ ☐☐

Answer: B (LEAVE A REPLY)

<https://learn.microsoft.com/en-us/azure/cosmos-db/how-to-setup-rbac#disable-local-auth>

NEW QUESTION: 81

□ □ □ □ □

○○○○ ○○○○○○ ○○○○ ○○○○ ○○○○ ○○ ○○ ○○ Azure Cosmos DB for NoSQL ○○○ ○○○ ○○○○○○. ○○ ○○○○○○ ○○ ○○ ○○○ ○○○○○○○○ ○○○○○○.

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.

☐ ☐☐☐☐☐☐☐ ☐☐☐ ☐☐ ☐☐ ☐☐☐☐ ☐☐☐☐.

☐ ☐☐☐☐☐☐☐ ☐☐ ☐☐ ☐☐ ☐☐☐ ☐☐? ☐☐☐☐ ☐☐ ☐☐☐☐ ☐☐ ☐☐☐☐.

□□: □□ □□□ 1□□□□.

Answer Area

Reporting:

- Bounded staleness
- Consistent prefix
- Eventual
- Session
- Strong

Purchasing:

- Bounded staleness
- Consistent prefix
- Eventual
- Session
- Strong

Fulfillment:

- Bounded staleness
- Consistent prefix
- Eventual
- Session
- Strong



Answer:

Reporting:

Purchasing:

Fulfillment:

Explanation:

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

NEW QUESTION: 82

□ □ □ □ □

□□ □□□ □□□□□□□ □□□□ □ □□ Azure Cosmos DB Core(SQL) API □□□ □□□ □□□□□. □□□ □□ □□ □□□ □□ □□□□□.

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

□□□ □□□□□□ □ □□ □ □□□□□ □□□ □□□□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

□□: □□ □□□ 1□□□□.

development:

	▼
Serverless capacity mode	
Provisioned throughput capacity mode and manual throughput	
Provisioned throughput capacity mode and autoscale throughput	

shipments:

	▼
Serverless capacity mode	
Provisioned throughput capacity mode and manual throughput	
Provisioned throughput capacity mode and autoscale throughput	

Answer:

development:		▼
	Serverless capacity mode	
	Provisioned throughput capacity mode and manual throughput	
	Provisioned throughput capacity mode and autoscale throughput	
shipments:		▼
	Serverless capacity mode	
	Provisioned throughput capacity mode and manual throughput	
	Provisioned throughput capacity mode and autoscale throughput	

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: 83

Azure Cosmos DB for NoSQL
 (DB1) , Container1

App1□□□ □□□ □□□□□ □□ □□□, □ □□ Container1□□ □□ □□□□ □□□□□□ □□ □□□.

DB1 is a global Azure Cosmos DB for NoSQL database. It has 10 partitions and 10 replicas. The database is configured with a composite index on the 'name' and 'age' properties. A client application is running on App1 and is making a query that filters on the 'name' property and orders the results by 'age'. The query is returning 429 errors (RequestRateTooLarge) and the client is experiencing high latency. What should you do to resolve this issue?

A. Increase the request rate limit for App1.

B. Add a single-property index on the 'name' property.

C. Add a single-property index on the 'age' property.

D. App1 is a global Azure Cosmos DB for NoSQL database. It has 10 partitions and 10 replicas. The database is configured with a composite index on the 'name' and 'age' properties. A client application is running on App1 and is making a query that filters on the 'name' property and orders the results by 'age'. The query is returning 429 errors (RequestRateTooLarge) and the client is experiencing high latency. What should you do to resolve this issue?

Answer: C (LEAVE A REPLY)

To minimize HTTP 429 (RequestRateTooLarge) errors while keeping costs low in a real-time, read/write global Azure Cosmos DB for NoSQL application, you should add a composite index. Reduces Read Costs (RUs): If your global application runs queries containing a WHERE clause with multiple properties or utilizes ORDER BY on multiple fields, standard single-property indexing is highly inefficient. A composite index allows Cosmos DB to resolve these complex queries instantly, slashing the consumed Request Units (RUs) per query. Mitigates 429 Errors: By dramatically lowering the RU consumption of your queries, your database stays well under its provisioned or autoscale throughput limits, effectively preventing rate-limiting (429) spikes. Keeps Overall Costs Low: The alternative to fixing query performance is scaling up the container's overall RU/s threshold, which scales up costs exponentially. A composite index targets the root cause safely without expensive over-provisioning. Reference: <https://learn.microsoft.com/en-us/answers/questions/5659021/indexing-strategy-for-cosmos-db>

NEW QUESTION: 84

Container1 is a global Azure Cosmos DB for NoSQL database. It has 20,000 items and 1,000 partitions. The database is configured with a composite index on the 'name' and 'age' properties. A client application is running on Container1 and is making a query that filters on the 'name' property and orders the results by 'age'. The query is returning 429 errors (RequestRateTooLarge) and the client is experiencing high latency. What should you do to resolve this issue?

A. Increase the request rate limit for Container1.

B. Add a single-property index on the 'name' property.

C. Add a single-property index on the 'age' property.

D. Container1 is a global Azure Cosmos DB for NoSQL database. It has 20,000 items and 1,000 partitions. The database is configured with a composite index on the 'name' and 'age' properties. A client application is running on Container1 and is making a query that filters on the 'name' property and orders the results by 'age'. The query is returning 429 errors (RequestRateTooLarge) and the client is experiencing high latency. What should you do to resolve this issue?

Answer: B (LEAVE A REPLY)

Too few items to evaluate the cost. Add more items. Reference: <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/how-to-query-container>

NEW QUESTION: 85

Azure Cosmos DB SQL API supports .NET Core 3.1 and .NET 5.0. The following code snippet shows how to create a database. Which method should you use to create the database?

A. BuildDatabaseAsync

B. CreateDatabaseAsync

C. CreateContainerAsync

D. CreateDatabaseAsync

Answer: D (LEAVE A REPLY)

A database is the logical container of items that are partitioned across containers. Either the CreateDatabaseAsync or CreateDatabaseIfNotExistsAsync method of the CosmosClient class can be used to create a database. Option A is incorrect. BuildDatabaseAsync is not a valid method to create the database. Option B is incorrect. CreateCosmosDatabase won't help in creating the database.

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

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

Reference:

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

NEW QUESTION: 86

You are using Azure Cosmos DB for NoSQL to store data. You want to monitor the database.

Which of the following metrics can you use to monitor the database?

A. Azure Monitor Provisioned Throughput

A. CPU usage

B. Azure Monitor Document Count

C. Azure Monitor Total Requests

D. Azure Monitor Region Removed

Answer: D (LEAVE A REPLY)

The recommended signal in Azure Monitor is Provisioned Throughput.

To receive an alert when requests target the database and exceed the available request units per second (RU/s), the Provisioned Throughput metric is used to track the number of request units (RUs) assigned to your resources.

Signal Selection Explained

When requests exceed the provisioned RU/s, Azure Cosmos DB returns a 429 (Too Many Requests) status code. Monitoring throughput helps you proactively manage capacity and costs.

Provisioned Throughput: This metric monitors the RU/s currently configured for a container or database. Using this signal allows you to set alert thresholds based on the maximum RUs your account can handle.

Total Requests (Alternative): While not one of the options, a common way to alert on throttling is to use the Total Requests metric and filter by the 429 status code.

Incorrect:

[Not A]

Document Count tracks the number of items stored; it does not reflect request traffic or RU usage.

[Not B]

Metadata Requests monitors operations like creating or listing containers, which are separate from standard data requests.

[Not D]

Region Removed is a control-plane event related to high availability and global distribution, not throughput consumption.

Reference:

<https://learn.microsoft.com/en-us/azure/cosmos-db/troubleshoot-request-rate-too-large>

NEW QUESTION: 87

You are using Azure Cosmos DB Core(SQL) API to store data. You want to monitor the database.

Which of the following metrics can you use to monitor the database?

A. Azure Synapse Apache Spark cosmos.olap CPU usage

B. Azure Synapse Apache Spark cosmos.olap Document Count

C. Azure Synapse Analytics SQL OPENROWSET CosmosDB Total Requests

D. Azure Synapse Link Apache Spark cosmos.olap CPU usage

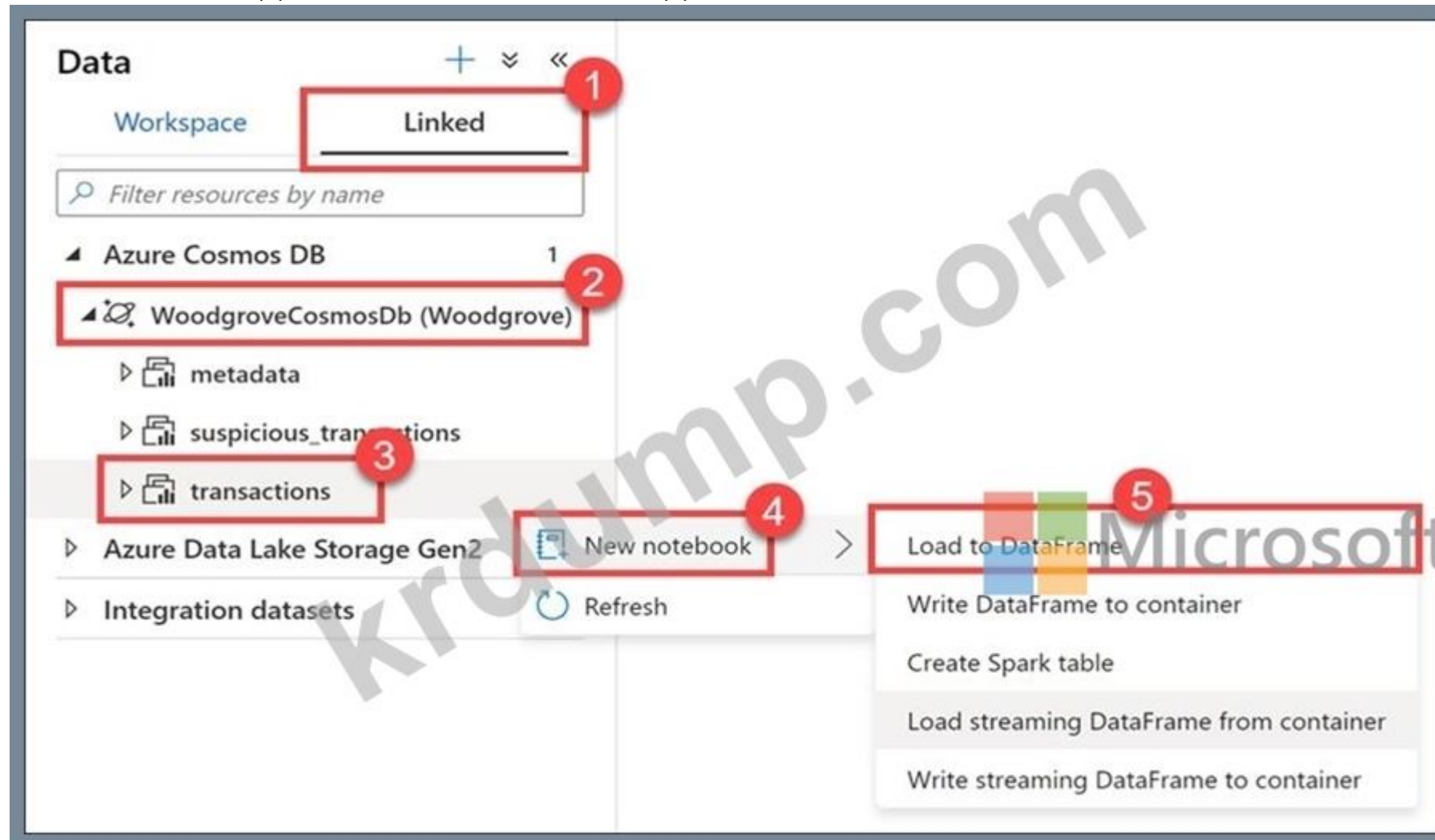
E. Azure Synapse ☐ Apache Spark ☐☐ cosmos.oltp ☐☐☐ ☐☐☐☐☐☐.

Answer: A,D (LEAVE A REPLY)

Explore analytical store with Apache Spark

1. Navigate to the Data hub.

2. Select the Linked tab (1), expand the Azure Cosmos DB group (if you don't see this, select the Refresh button above), then expand the WoodgroveCosmosDb account (2). Right-click on the transactions container (3), select New notebook (4), then select Load to DataFrame (5).



3. In the generated code within Cell 1 (3), notice that the `spark.read` format is set to `cosmos.olap`.

This instructs Synapse Link to use the container's analytical store. If we wanted to connect to the transactional store, like to read from the change feed or write to the container, we'd use `cosmos.oltp` instead.

Database operations against the service such as reads, writes, and queries consume some number of request units (RUs).

In a given second, if the operations consume more than the provisioned request units, Azure Cosmos DB will return a 429 exception. Each second, the number of request units available to use is reset.

There are different error messages that correspond to different types of 429 exceptions:

(D) *-> Request rate is large. More Request Units may be needed, so no changes were made.

The request didn't complete due to a high rate of metadata requests.

(E) *-> The request didn't complete due to a transient service error.

Reference:

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/troubleshoot-request-rate-too-large>

NEW QUESTION: 89

□□ □□ □□□ □□ Azure Cosmos DB for NoSQL □□□ □□□□.

□□□□□□□ □□□□ □□ □□□ □□ □□ □□□ □□ □□(RU/s)□ □□□ □□ □□□ □□□ □□□.

□□ Azure Monitor □□□ □□□□ □□□?

A. □□□□□ □□

B. □□ □

C. □□□ □□□

D. □□ □□□

Answer: A ([LEAVE A REPLY](#))

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: 90

□□□□ 1□□□ □□□□□ □□□ Azure Cosmos DB □□□□□□□ □□□□. □□□□ 1□ □□ 20,000RU/s□ □□□□ □□□ □□ 240GB□ □□□□ □□□□ □□□□.

□□ □□□□ □□□□ container1□ □□□ □□□□ □□□.

□ RU/s □ □ □ □ □ ?

A. 240

B. 4.000

C. 20,000

D. 24000

Answer: C (LEAVE A REPLY)

NEW QUESTION: 91

□□ □□ □□□ □□ □□ □ □□ □□ Azure Cosmos DB for NoSQL □□□ □□□ □□□□□.

□□□ □□ □□□ □□□ □□□□ □□□. □□□□ □□ □□ □□□ □□□□ □□□.

- □□ □□□□ □□□ □□□□□ □□□□□ □□□.

□□ 3□□ □□□.

- □□ □□□□ □□□□ □□□ □□□□□ □□ □□□ □□ □□□ □□□ □□□□ □□□.

- □□ □ □□ □□□□ □□□□□ □□□.

□□ □□□ □□□ □□□□ □□□?

A. □□

B. ☐☐☐ ☐☐

C. ☐☐☐ ☐☐☐

D. □□

Answer: ([SHOW ANSWER](#))

DP-420-KR ☐☐ ☐☐☐ ☐☐☐☐☐ ☐☐ DumpTop ☐☐ ☐☐☐☐ ☐☐☐ DP-420-KR ☐☐! DumpTop ☐ ☐☐ **DP-420-KR** ☐☐ ☐☐☐ ☐☐☐☐☐☐, DumpTop DP-420-KR ☐☐ ☐☐☐ ☐☐☐☐☐☐☐☐ ☐☐☐ ☐☐☐☐☐☐☐. ☐☐☐☐ ☐☐☐ ☐☐☐☐☐ ☐☐ DumpTop DP-420-KR ☐☐☐ ☐☐☐☐☐. <https://www.dumptop.com/Microsoft/DP-420-KR-dump.html> (281 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 92

Azure Cosmos DB for NoSQL ☐☐☐ container1☐☐☐ ☐☐☐☐☐ ☐☐☐☐. container1☐ ☐☐☐ 3☐☐☐ ☐☐☐☐☐.

1. Create a new function in the Azure Functions portal.

□□1□ □□□□□ □□□, □□ upsert □□□□ □□□□ □□ □□□□ □□ □□ □□□□□.

```
function1 upsert 1 upsert
```

function1 Function.json □□□□ □□ □□□ □□□□ □□□?

A. □□□□□□□

B. ☐☐ ☐☐☐ ☐☐☐

C. ☐☐☐ ☐☐ ☐☐ ☐

D. ☐ ☐ ☐ ☐ ☐ ☐

Answer: D (LEAVE A REPLY)

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: 93

account1_____ Azure Cosmos DB for NoSQL _____.

account1_____.

account1_____ . _____ . _____ .

_____ ? _____ .

____: _____ 1_____.

Answer Area

Connectivity mode:

Direct mode over TCP

Gateway mode with a dedicated gateway

Gateway mode with a standard gateway

Consistency level:

Consistent Prefix

Session

Strong

Microsoft

Answer:

Answer Area

Connectivity mode:

- Direct mode over TCP
- Gateway mode with a dedicated gateway
- Gateway mode with a standard gateway

Consistency level:

Consistent Prefix

NEW QUESTION: 94

Azure Cosmos DB for NoSQL □□□ □□□□.

□□□□□ □□□ □□□□□□ □□ □□□ □□□□□ □□□□.

□□ □□□ □□□□ □□ Azure □□□ □□□□□.

Azure ☐☐☐ ☐☐☐ ☐☐☐☐☐?

A. □□□□□ □□□ □□□ □□□ □□□ □□ □□□ □□□□□.

B. □□□ □□ □□□□□ □□□ □□□ □□□ □□□□□.

C. ☐☐ ☐☐☐ ☐☐☐☐☐ ☐☐☐ ☐☐ ☐☐

D. ☐☐☐☐☐☐ ☐☐☐ ☐☐ ☐☐

Answer: ([SHOW ANSWER](#))

All properties are sent to the change feed. This is testable by creating an Azure Function with CosmosDB input trigger.

NEW QUESTION: 95

Container1 Azure Cosmos DB for NoSQL

□□ □□ □□□□ □□□□□.

Container1

☐☐☐ ☐☐☐☐ ☐☐☐.

□□□ □□□□ □□□?

A. ☐☐ ☐☐ ☐☐☐☐

B. ☐☐ ☐☐

C. ☐☐☐ ☐☐ ☐☐

D. ☐☐ ☐☐☐☐☐

Answer: A ([LEAVE A REPLY](#))

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: 96

Azure Cosmos DB for NoSQL ☐☐☐ ☐☐☐☐☐ ☐☐☐.

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ .

□□□□□□ □□ □□□□ □□□□□□ □□ □□□□□□ □□□□□ □□□ □□□□□□. □□ □□□□ Azure Cosmos DB SDK□ □□ □□ □□□□□□ □□□□□ □□□□□□. □□ □□□□□□ □□□□□□ □□□□□ □□□□.

□□ □ □□ □□□ □□ □□□? □ □□□ □□□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

- A.** □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ .
- B.** □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ .
- C.** □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ .
- D.** □ .
- E.** □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ .
- F.** □ .

Answer: A,C,F (LEAVE A REPLY)

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: 97

How can I connect my application to Azure Cosmos DB using the Azure Cosmos DB SDK for Java?

- A.** SQL
B. MongoDB
C. ☐☐☐☐
D. ☐☐☐
E. ☐

Answer: A,E (LEAVE A REPLY)

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

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

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

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

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

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

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

Reference:

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

NEW QUESTION: 98

□□□ □□

Azure Cosmos DB for NoSQL □□□ □□□□ □□□□□□□ □□□□ □□□□.

□□ □□□□ □□/□□ □□ □□□ □□ □□ □□ □□□□ □□□□. □□□ □□□□ □□ □□ □□(failover)□ □□ □□□□ □□□□.

□□ □□□□ □□□ □□ □□ □□□□ □□□□. (□ □□□ □□□□□□ □□□□ □□□□.)

```

01 {
02   "connectionStrings": [
03     {
04       "connectionString":
05       "AccountEndpoint=https://constosodbaccount.documents.azure.com:443/;
06       AccountKey=MwUgRnGti4vErT2rfPPFdTFFyI9KyI9KbelRPGv7OQdHo6VZ2i45TcJzrd4J80zYxrEATzyZh0mlnJaNFA==;",
07       "description": "Primary SQL Connection String"
08     },
09     {
10       "connectionString":
11       "AccountEndpoint=https://constosodbaccount.documents.azure.com:443/;
12       AccountKey=gfThRnGti4vErT2rfPPFdTFFyI43529KbelRPGv7OQdHo6VZ2i45TcJzrd4J80zYxrEATzyZh0mlnJaNFA==;",
13       "description": "Secondary SQL Connection String"
14     },
15     {
16       "connectionString":
17       "AccountEndpoint=https://constosodbaccount.documents.azure.com:443/;
18       AccountKey=WGykBc1PHJoos6MdErT2rfPPFx9yI9KbelRPGv7OQlIQwQNxq6QdOXjxgyLLebXBp8uJu7FyJy3Uv1vuK2A==;",
19       "description": "Primary Read-Only SQL Connection String"
20     },
21     {
22       "connectionString":
23       "AccountEndpoint=https://constosodbaccount.documents.azure.com:443/;
24       AccountKey=k2DZI0oY4Jc7QeUJqVGH3csda6EyI9KbelRPGv7OQErT2rfPPFtbwTPfKAgl9zVxC0MDNn8xPpQrednVYcQ==;",
25       "description": "Secondary Read-Only SQL Connection String"
26     }
27   ]
28 }

```

□□ □ □□□ □□, □□□ □□□□ '□' □ □□□□, □□□ □□□ '□□□□' □ □□□□□□□.

□□: □□ □□□ 1□□□□□.

Answer Area

Microsoft

Statements

	Yes	No
If the primary write region fails, applications that write to the database must use a different connection string to continue to use the service	☐	☐
The Primary Read-Only SQL Connection String and the Secondary Read-Only SQL Connection String will connect to different regions from an application running in the East US Azure region	☐	☐
Applications can choose from which region they read by setting the PreferredLocations property within their connection properties	☐	☐

Answer:

Answer Area

Microsoft

Statements

	Yes	No
If the primary write region fails, applications that write to the database must use a different connection string to continue to use the service	☐	☒
The Primary Read-Only SQL Connection String and the Secondary Read-Only SQL Connection String will connect to different regions from an application running in the East US Azure region	☐	☒
Applications can choose from which region they read by setting the PreferredLocations property within their connection properties	☒	☐

Explanation:

Box 1: No
The same connection string can still be used.
Azure Cosmos DB Core read write region connection strings
Azure Cosmos DB "primary write region" failover "connection string"

Box 2: No
The AccountEndpoint for both are the same.

Box 3: Yes
The ConnectionPolicy.PreferredLocations property gets and sets the preferred locations (regions) for geo-replicated database accounts in the Azure Cosmos DB service. For example, "East US" as the preferred location.

When EnableEndpointDiscovery is true and the value of this property is non-empty, the SDK uses the locations in the collection in the order they are specified to perform operations, otherwise if the value of this property is not specified, the SDK uses the write region as the preferred location for all operations.

Reference:

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.documents.client.connectionpolicy.preferredlocations>

NEW QUESTION: 99

Container1 is an Azure Cosmos DB for NoSQL container.

Azure Cosmos DB .NET SDK is used to query Container1.

Which of the following is true?

Options: A. The SDK uses the write region as the preferred location for all operations. B. The SDK uses the locations in the collection in the order they are specified to perform operations. C. The SDK uses the preferred location for all operations. D. The SDK uses the write region as the preferred location for all operations.

Answer: B

A. IDocumentQuery

B. FeedResponse

C. StoredProcedureResponse

D. ResourceResponse

Answer: A,D (LEAVE A REPLY)

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: 100

Options: A. The SDK uses the write region as the preferred location for all operations. B. The SDK uses the locations in the collection in the order they are specified to perform operations. C. The SDK uses the preferred location for all operations. D. The SDK uses the write region as the preferred location for all operations.

Account1 is an Azure Cosmos DB for NoSQL container. Vault1 is an Azure Key Vault.

Which of the following is true?

Options: A. The SDK uses the write region as the preferred location for all operations. B. The SDK uses the locations in the collection in the order they are specified to perform operations. C. The SDK uses the preferred location for all operations. D. The SDK uses the write region as the preferred location for all operations.

Answer: B

A. IDocumentQuery

B. FeedResponse

C. StoredProcedureResponse

D. ResourceResponse

Answer Area



For Vault1:

Create or select a key and record its key name.
Create or select a key and record its key version.
Enable automatic key rotation and copy the latest version of the URI.
Generate a new key and copy its secret for Account1.

For Account1:

Configure Account1 to rotate the keys automatically.
From the Encryption settings, enter the key name from Vault1.
From the Encryption settings, enter the key version from Vault1.
Manually upload the key material from Vault1.

Answer:

Answer Area

Microsoft

For Vault1:

Create or select a key and record its key name.

Create or select a key and record its key version.

Enable automatic key rotation and copy the latest version of the URI.

Generate a new key and copy its secret for Account1.

For Account1:

Configure Account1 to rotate the keys automatically.

From the Encryption settings, enter the key name from Vault1.

From the Encryption settings, enter the key version from Vault1.

Manually upload the key material from Vault1.

NEW QUESTION: 101

□□□ □□
Azure Cosmos DB for NoSQL □□□□□□(□□: db1)□ □□□□.
□□ □□□ □□□□□.

Database database = client.GetDatabase("db1");

```
ContainerProperties properties = new()  
{  
    Id = "customers",  
    PartitionKeyPath = "/customerId",  
    DefaultTimeToLive = -1  
};
```

```
Container container = await database  
    .CreateContainerAsync(properties);
```

Which of the following statements are true?
Select all that apply.

Answer Area

Statements

Items inserted into the customers container without a ttl value will never expire.

Yes

☐

No

☐

All Items inserted into the customers container must include a customerId field.

☐☐

Items inserted into the customers container that have a TTL value of 12 will expire in 12 days.

☐☐

Answer:

Answer Area



Statements

Items inserted into the customers container without a ttl value will never expire.

Yes

☒

No

☐

All Items inserted into the customers container must include a customerId field.

☐☒

Items inserted into the customers container that have a TTL value of 12 will expire in 12 days.

☐☒

NEW QUESTION: 102

□□□ □□

□□ □□ Azure Cosmos DB for NoSQL □□□□□ □□ □□□□.

Name	DefaultTimeToLive
container1	-1
container2	null
container3	60

□□□ □□ □□ □□□ □□□□ □□□□ □□□□.

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

□ □□□ □□□□□ □□□□□□□? □□□ □□□□□ □□□□□ □□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

Answer Area

Item1:

	▼
Never	
10 seconds	
60 seconds	
10 minutes	
60 minutes	

Item2:

	▼
Never	
10 seconds	
60 seconds	
10 minutes	
60 minutes	



Microsoft

Item3:

	▼
Never	
10 seconds	
60 seconds	
10 minutes	
60 minutes	

Answer:

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	

Explanation:

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

NEW QUESTION: 103

□□□ □□□ IoT □□□□ □□ □□□□ □□ □□ □□□□ □□□□ □□ □□□□ □□□□.

□□□ □ □□ □□, □ □□□ □ □□□□ □□□□ □□ □□□□ □□ □□□□ □□□□□ □□□□□.

□□□ □□□□□ □□ □ □□ □□□□□ □□□□ □□□, □□□ □□□ □□□□□ □ □□□□ □□ □□ □□□□□ □□ □□ □□□□□□.

□□□□□ □□□ □□□□□□?

A. deviceid □□□ □□ □□ □□□□ □□□□ □□□ □□□□, □□ □□□ □□□□□.

'type' □□□ 'device' □□ □□□□□. □□ devicemetricsid □□□ □□□□ □□□ □ □□□ □□□□ □□ □□□ □□□ □□□□□.

B. deviceid □□□ □□ □□ □□□□ □□□□ □□□ □□□□□. □□ □□ devicemetricsid □□□ □□ □□□ □□□□ □□□□ □ □□□ □□□□ □□□ □□□□□.

C. deviceid □□□ □□ □□ □□□□ □□□□ □□□ □□□□, □□ □□□ □□□□□.

'type' □□□ 'device' □□ □□□□□. devicemetricsid □ deviceid □□□ □□□□ □ □□□ □□□□ □□ □ □□□ □□□, 'type'□□□ □□□ 'devicemetrics' □□ □□□□□.

D. □□ □ □□ □□ □□□□.

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: 104

Which of the following is a valid column name for a table in Azure Cosmos DB for NoSQL? (Select all that apply.)

Name	Column
Author 	authorId (primary key)
	fullname
	address
	contactinfo
Book	bookId (primary key)
	isbn
	title
	genre
BookauthorInk	authorId (foreign key)
	bookId (foreign key)

Which of the following is a valid column name for a table in Azure Cosmos DB for NoSQL? (Select all that apply.)

Which of the following is a valid column name for a table in Azure Cosmos DB for NoSQL? (Select all that apply.)

- A. authorId (primary key) is a valid column name. It is a valid column name, and it is a valid column name.
- B. bookId, isbn, title, genre are valid column names. They are valid column names.
- C. authorId is a valid column name. It is a valid column name.
- D. authorId (foreign key) is a valid column name. It is a valid column name.

Answer: A (LEAVE A REPLY)

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/modeling-data#hybrid-data-models>

NEW QUESTION: 105

Which of the following is a valid column name for a table in Azure Cosmos DB for NoSQL? (Select all that apply.)

Which of the following is a valid column name for a table in Azure Cosmos DB for NoSQL? (Select all that apply.)

- A. authorId (primary key) is a valid column name. It is a valid column name.
- B. bookId, isbn, title, genre are valid column names. They are valid column names.
- C. authorId is a valid column name. It is a valid column name.
- D. authorId (foreign key) is a valid column name. It is a valid column name.

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: 106

□□□ □□

Azure Cosmos DB for NoSQL □□□□□□(DB1)□ □□□, □ □□□□□□□ Container1□□□ □□□□□ □□□□□□. Container1□ □□□ □□ /pk1□□□.

DB1□ □□□□ .NET □□□□□□□□ □□□□.

□□□ □□□ □□ □□(RU) □□□ □□□□ □□ □□ □□□ □□□□ □□□.

C# □□□ □□□ □□□□ □□□? □□□□ □□ □□□□ □□□ □□□ □□□□□□.

□□: □□ □□□ 1□□□□.

Answer Area

```
var container = cosmosClient.GetContainer("db1", "container1");  
var item = new { id = "id1", pk = "pk1" };  
var response = await container.
```

CreateItemAsync<dynamic>(item);
DeleteContainerAsync();
GetItemQueryIterator<dynamic>(new QueryDefinition("SELECT * FROM c"));

```
var ruCharge = response.
```

Diagnostics.GetClientElapsedTime().TotalMilliseconds;
RequestCharge;
Resource.RequestCharge;

Answer:

Answer Area

```
var container = cosmosClient.GetContainer("db1", "container1");  
var item = new { id = "id1", pk = "pk1" };  
var response = await container.
```

CreateItemAsync<dynamic>(item);
DeleteContainerAsync();
GetItemQueryIterator<dynamic>(new QueryDefinition("SELECT * FROM c"));

```
var ruCharge = response.
```

Diagnostics.GetClientElapsedTime().TotalMilliseconds;
RequestCharge;
Resource.RequestCharge;

DP-420-KR                        

NEW QUESTION: 107

account1□□□ Azure Cosmos DB for NoSQL □□□ □□□, □ □□□ App1□□□ □□□□□□□ □□□□□. App1□ □□□ □□□ □□□ □□□ □□□□□.

account1□ .

App1□ □□ □□□ □□□ □ □□□ □□ □□□.

App1□ □□ □□ □ □□ □□□ □□□□ □□□? □ □□□ □□ □□□ □□□ □□□□□.

□□: □□ □□□ 1□□□□.

A. ☐☐☐ ☐☐ ☐☐☐ ☐☐☐ ☐☐☐☐☐☐

B. □□ □□□□□□ http://account1.documents.azure.com□□ □□□□□.

C. ☐☐ ☐☐☐☐☐☐ <http://account1.sqlx.cosmos.azure.com> ☐☐ ☐☐☐☐.

D. □□ □□□ □□ □□□ □□□□□.

E. □□□ □□□ □□□ □□□□□.

Answer: A,C (LEAVE A REPLY)

A: You must adjust the request consistency to session or eventual. If not, the request will always bypass the integrated cache.

C: Modify your application's connection string to use the new dedicated gateway endpoint.

All dedicated gateway connection strings follow the same pattern. Remove `documents.azure.com` from your original connection string and replace it with `sqlx.cosmos.azure.com`. A dedicated gateway will always have the same connection string, even if you remove and reprovision it.

Reference:

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

NEW QUESTION: 108

□ □ □ □ □

account1□□□ □□□ Azure Cosmos DB for NoSQL □□□ db1□□□ □□□ □□□□□□□ □□□□.

db1 □□□□□□□□ □□ □□□□ □□ 4,000 □□ □□(RU/s)□□□.

Azure CLI `az storage blob upload-batch --source-dir ./ --destination-path db1 --overwrite true --progress`

□□ □□□□ □□ 4,000 RU/s, □□ 40,000 RU/s □□□□ □□□□ □□□□.

CLI □□ □□□ □□□□ □□□? □□□□ □□ □□□□ □□□ □□□ □□□□□□.

□□: □□ □□□ 1□□□□.

Answer Area

```
az cosmosdb sql database throughput
```

migrate
show
update

```
-a "account1" \
```

```
-g "cosmosdbrg" \
```

```
-n "db1" \
```

```
-t 'autoscale'
```

```
az cosmosdb sql database throughput
```

migrate
show
update

```
-a "account1" \
```

```
-g "cosmosdbrg" \
```

```
-n "db1" \
```

```
--throughput
```

400
4000
40000

 Microsoft

Answer:

Answer Area

az cosmosdb sql database throughput

-a "account1" \

-g "cosmosdbrg" \

-n "db1" \

-t 'autoscale'

migrate

show

update

az cosmosdb sql database throughput

-a "account1" \

-g "cosmosdbrg" \

-n "db1" \

--throughput

400

4000

40000

Explanation:

Box 1: migrate

The az cosmosdb sql database throughput migrate command migrates the throughput of the SQL database between autoscale and manually provisioned.

Syntax: az cosmosdb sql database throughput migrate

- account-name
- name
- resource-group
- throughput-type {autoscale, manual}

Box 2: update

The az cosmosdb sql database throughput update command updates the throughput of the SQL database under an Azure Cosmos DB account.

Syntax: az cosmosdb sql database throughput update

- account-name
- name
- resource-group
- [--max-throughput]
- [--throughput]

Box 3: 4000

Specify the throughput.

Parameter --throughput -

The throughput of SQL database (RU/s).

Note: Example migration from standard (manual) provisioned throughput to autoscale: Suppose you have a container with 10,000 RU/s manual provisioned throughput, and 25 GB of storage.

When you enable autoscale, the initial autoscale max RU/s will be: 10,000 RU/s, which will scale between 1000 - 10,000 RU/s.

Note 2: Parameter --max-throughput

The maximum throughput resource can scale to (RU/s). Provided when the resource is autoscale enabled. The minimum value can be 4000 (RU/s).

Reference:

<https://docs.microsoft.com/en-us/cli/azure/cosmosdb/sql/database/throughput>

NEW QUESTION: 109

Which of the following Azure Cosmos DB configurations cannot be used in conjunction with Azure Private Link?

- A. Integrated cache
- B. Azure Private Link
- C. Azure Cosmos DB Core(SQL) API
- D. Azure Cosmos DB NoSQL API

Answer: B (LEAVE A REPLY)

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: 110

Which of the following is a valid Azure Cosmos DB Core(SQL) API container name?

Which of the following is a valid Azure Cosmos DB Core(SQL) API database name?

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

Which of the following is a valid Azure Cosmos DB Core(SQL) API container name?

- cn1: container name, db1: database name, Fn1: function name
- cn2: container name, db1: database name, Fn2: function name
- cn3: container name, db1: database name, Fn3: function name

Which of the following is a valid Azure Cosmos DB Core(SQL) API database name?

- cn1: container name, item1: item name, db1: database name
- cn2: container name, item2: item name, db1: database name

cn3: container name, item3: item name, db1: database name, (Time to Live) 3,600 seconds. The container name must be unique within the database, and the item name must be unique within the container.

Answer: A, B, C, D

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	☐	☐

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	☒	☐

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: 111

Azure db1 is an Azure Cosmos DB for NoSQL database. User1 is a Microsoft Entra ID user.

User1 is connected to db1.

- User1 is connected to db1.
- User1 is connected to db1.
- User1 is connected to db1.

User1 is connected to db1.

A. db1 is a database.

B. □□□□ DB □□ □□□ □□□

C. ☐☐☐☐ DB ☐☐ ☐☐☐ ☐☐

D. □□□□ DB □□□

Answer: D (LEAVE A REPLY)

To ensure the user can only manage the database without accessing connection strings or stored data, you should assign the Cosmos DB Operator role.

Role Details and Permissions

This role is specifically designed for the control plane, allowing administrative tasks while strictly isolating data-plane access.

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: 112

□□□□ □□ □□□□□ □□□□ □□□, □ □□□ 4□□□ □□ □□□□ □□□□ □□□, 1, 2□□□□ □□□ □□ □□□□ □□, 3, 4□□□□ □□ □□ □□□□ □□□□.

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ 50,000 RU/s ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ countryId ☐ ☐ ☐ ☐ ☐ ☐.

□□ □ □□ □□□ □□□□□?

A. ☐ ☐ ☐ ☐ ☐

B. □ □□□ □□ □□□ □ □□□□ □□□□□.

C. □ □□□ □□ countryId□ □□□□ □ □□□ □□□ □□□□□.

D. □ .

Answer: D (LEAVE A REPLY)

NEW QUESTION: 113

Azure Cosmos DB for NoSQL □□□ □□□□.

□□ □□□ □□□□ □□ □□ □□□ Log Analytics □□ □□□□ □□□□□.

□□ □ □□□□ □□ □□ □□□□□□ □□ □□(RU/s) □□ □□□□□□ □□□□ □□□.

□ □ □ □ □ □ □ □ □ □ □ □ □ .

AzureDiagnostics

| □□□□□ "ControlPlaneRequests" □ □ □

□□□ □□□ □□□□ □□□?

A. | OperationName□ "AccountUpdateComplete"□ □□□□ □□

B. | OperationName "SqlContainersCreate" | | | | |

C. | OperationName "MongoCollectionsThroughputUpdate" | | | | |

D. | OperationName "SqlContainersThroughputUpdate" | | | | |

Answer: D (LEAVE A REPLY)

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.

Answer Area

Scenario 1:

One ARM template deployment in complete mode

One ARM template deployment in incremental mode

Two ARM template deployments in incremental mode

Scenario 2:

Azure PowerShell

One ARM template deployment in complete mode

One ARM template deployment in incremental mode

Answer:

Scenario 1:

One ARM template deployment in complete mode

One ARM template deployment in incremental mode

Two ARM template deployments in incremental mode

Scenario 2:

Azure PowerShell

One ARM template deployment in complete mode

One ARM template deployment in incremental mode

Azure Cosmos DB uses spatial indexes for geospatial objects, such as points, polygons, and line strings. Which of the following is not a supported spatial index type?

- A. Range index
- B. Composite index
- C. Spatial index
- D. Spatial index

Answer: C (LEAVE A REPLY)

Spatial Indexes are used for geospatial objects. Currently, it supports Points, Polygons, MultiPolygons, and LineStrings. Spatial indexes can also be used on GeoJSON objects. Option A is incorrect. Range indexes are not used on geospatial objects. Option B is incorrect. Use composite indexes when you need to enhance the efficiency of queries that execute operations on several fields. Option C is correct. Spatial indexes are utilized for geospatial objects. Option D is incorrect. The spatial index is the right answer.

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

NEW QUESTION: 116

.NET SDK v3 uses the Azure Cosmos DB SDK to connect to the database. WestUS2 is the region where the database is replicated. Which of the following is not a supported region?

- A. WestUS2
- B. EastUS2
- C. EastUS
- D. WestUS

Answer: (SHOW ANSWER)

In .NET SDK v3, if you want to enable multi-region writes in your application, set ApplicationRegion to the region where the application is being deployed and Cosmos DB is replicated.
<https://docs.microsoft.com/bs-latn-ba/azure/cosmos-db/sql/how-to-multi-master>

NEW QUESTION: 117

Which of the following is not a supported index type for Azure Cosmos DB for NoSQL? - Range index - Composite index - Spatial index - Spatial index

Answer Area

Storage mechanism:

Delegate
Host
Lease container
Monitored container

Code that implements business logic:

Delegate
Host
Lease container
Monitored container

Answer:

Answer Area

Storage mechanism:

Delegate
Host
Lease container
Monitored container

Code that implements business logic:

Delegate
Host
Lease container
Monitored container

NEW QUESTION: 118

Container1 is a Microsoft Azure Cosmos DB for NoSQL container. Container1 is configured with the following properties:

Container1 is configured with the following properties:



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

Answer Area

"move",	"from":
"replace",	"path":
"set",	"value":

Answer:

Answer Area

```
[
  { "op": 

|            |   |
|------------|---|
|            | ▼ |
| "move",    |   |
| "replace", |   |
| "set",     |   |



|          |   |
|----------|---|
|          | ▼ |
| "from":  |   |
| "path":  |   |
| "value": |   |

 "/color", "path": "/inventory/color" }
]
```

NEW QUESTION: 119

□ □ □ □ □ □ □ □

Azure search1 Azure AI Azure Cosmos DB for NoSQL Container1

```
search1  Container1  
```

□□ □ □□ □□□ □□□□ □□□□ □□□? □□□ □□□□□, □□ □□□□ □□□ □□□ □□ □□□ □□ □□□ □□□□ □□□□□.

- □□ □□ □□ □□ □□ □□□□.
- □□□ □□□□ □□□□□.
- □□□ □□□ □□□□ □□□?
- A. □□
- B. □□□ □□
- C. □□
- D. □□□ □□□

Answer: D ([LEAVE A REPLY](#))

<https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels>

NEW QUESTION: 121

□□□ □□

□□□ □□□□ □□ □□ □□□ □□ Azure Cosmos DB for NoSQL □□□ □□□□ □□□□.

□□ Azure CLI □□□□□ □□□□□.

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

□□ □ □□□ □□, □□□ □□□□ '□'□ □□□□, □□□ □□□ '□□□'□ □□□□□□.

□□: □□ □□□ 1□□□□.

Statements

After running the script, there will be an instance of Azure Cosmos DB in North Europe that is writable

Yes	No
☐	☐

After running the script, the Azure Cosmos DB instance in West Europe will be writable

☐	☐
-----------------------	-----------------------

The cost of the Azure Cosmos DB account is unaffected by running the script

☐	☐
-----------------------	-----------------------

Answer:

- A. 0
- B. 1
- C. 2
- D. 3

Answer: A (LEAVE A REPLY)

Based on the configuration, there would be 0 conflicts. Azure Cosmos DB unique keys are enforced within a specific logical partition (defined by the /ID partition key). The unique key constraint is defined on a combination of User and EmailAddress. Each item is analyzed for uniqueness only against other items within the same ID (Company1 or Company2).

Analysis of Actions:

{ID Company1, user1, user1@contoso.com}: Added (First in Company1).
{ID Company1, user2, U1@contoso.com}: Added (New combination in Company1). Note: Unique keys are case-sensitive, so user1@contoso.com and U1@contoso.com are different.
{ID Company1, user2, user2@outlook.com}: Added (New combination in Company1).
{ID Company2, user1, user1@litwareinc.com.com}: Added (First in Company2).
{ID Company2, user2, user1@outlook.com}: Added (New combination in Company2).
All pairs of (User, EmailAddress) within their respective /ID partitions are unique.

Reference:
<https://learn.microsoft.com/en-us/azure/cosmos-db/unique-keys>

NEW QUESTION: 123

□□□ □□
disableKeyBasedMetadataWriteAccess □□□ □□□□ account1□□□□ Azure Cosmos DB Core(SQL) API □□□ □□□□.
□□□ DevUser1□□□ □□□□ account1□ □□□□□ □□□□ □ □□□ App1□□□ □□ □□□□ □□□□.
DevUser1□ Azure Active Directory(Azure AD) □□□□ □□□ □□ □□□ □□□ □□□□.
DevUser1□ App1□ □□□□ account1□ □□□□□ □□□ □ □□□ □□ □□□.
□□□ □□ □□□? □□□□□ □□□□□ □□□ □□□ □□□□□.
□□: □□ □□□ 1□□□□□.

Grant permissions to create containers by using:

	▼
Account keys	
Resource tokens	
Role-based access control (RBAC)	

Create containers by using the:

	▼
Azure AD Graph API	
Azure Resource Manager API	
SQL (Core) API	

Answer:

Grant permissions to create containers by using:

Account keys

Resource tokens

Role-based access control (RBAC)

Create containers by using the:

Azure AD Graph API

Azure Resource Manager API

SQL (Core) API

Explanation:
Box 1: Role-based access control (RBAC)
Box 2: Azure Resource Manager API
You can use Azure Resource Manager to help deploy and manage your Azure Cosmos DB accounts, databases, and containers.
Reference:
<https://docs.microsoft.com/en-us/azure/cosmos-db/secure-access-to-data>
<https://docs.microsoft.com/en-us/rest/api/resources/>

NEW QUESTION: 124

Q: You are developing an application that uses the Azure Cosmos DB SDK. You need to configure the SDK to use the Last-Writer-Wins conflict resolution policy. Which of the following options should you use to configure the SDK?

A. `ConflictResolutionMode.Custom`

B. `ConflictResolutionMode.LastWriterWins`

Answer: (SHOW ANSWER)

There are three ways for conflict resolution with the SDK:

1. Last-Writer-Wins: Conflicts do NOT show up in conflict feed
2. Custom Conflict Resolution Policy with Stored Proc: Conflicts do NOT show up in conflict feed unless there's an error in your stored procedure
3. Custom Conflict Resolution Policy: Conflicts DO show up in conflict feed.

NEW QUESTION: 125

Which Azure CLI command can be used to change the failover priority of the db1 region in an Azure Cosmos DB for NoSQL database?

- A. az cosmosdb failover-priority-change
- B. az cosmosdb failover-priority-change
- C. az cosmosdb failover-priority-change
- D. az cosmosdb sql failover-priority-change

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: 126

Azure Cosmos DB offers five well-defined levels. From strongest to weakest, the levels are:

SELECT consistency (Consistent Prefix)

SELECT consistency (Consistent Prefix)

SELECT consistency (Consistent Prefix)

A. Strong

B. Bounded staleness

C. Session

D. Consistent prefix

Answer: D (LEAVE A REPLY)

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/consistency-levels>

NEW QUESTION: 127

Azure Cosmos DB for NoSQL □□□ □□□□ □□□ □□□ □□□□ Azure Data Factory □□□ □□□ □□□□ □□□□. □ □□□ □□□ 2,000□□ Apache Spark □□□□ □□□□□.

□□□□ □□□□□□ □ Spark □□□□ □□□ □□□□ □□□ □□□□ □□ □□□.

□□ □□□ □□□ □□□□ □□□?

A. ☐ ☐ ☐

B. ☐☐ ☐☐☐ ☐☐

C. ☐ ☐ ☐ ☐

D. ☐ ☐ ☐ ☐

Answer: C (LEAVE A REPLY)

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: 128

Azure Cosmos DB for NoSQL □□□ □□□ Azure □□□ □□□□. □□ □□□ Container1□□□ □□□ □□□□□ □□□□□□.

□□ 4□ □□ □□ □□□ Container1□ □□□□□□□.

Container1□ □□□ □□ □□□□ □□□ □□□□ □□□. □□ □□□ □□□□ □□□ □□□□□ □□□.

□□□ □□ □□ □□□?

A. □□□ □□ □□□□□ □□□□□.

B. □□□□□ □□ □□□ □□□□□□ □□□□□.

C. ☐☐☐ Azure Cosmos DB for NoSQL ☐☐☐ ☐☐☐☐.

D. Container1 ☐ ☐ Azure Cosmos DB for NoSQL ☐ ☐ ☐ ☐ ☐.

Answer: B (LEAVE A REPLY)

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: 129

□□ □□□□□□ □□□□ □□□□□ □□□ Azure Cosmos DB □□□ □□□□ □□ □ □□□□□□. □□ □ □□ □□□□□□□ □□ □□□□□□□ □□ □□□□□□?

A. □□ □□□□□□ SQL API □□□□□ □□□□□.

B. IP [] [] [] [] [] [] [] [] Azure Cosmos DB [] [] [] [] [] [] [] [].

C. Azure Cosmos DB □□□ □□ □□□□(Vnet)□ □□ □□□□□□ □□□□□□ □ □□□□.

D. RBAC(☐☐ ☐☐ ☐☐ ☐☐)☐ ☐☐☐☐ ☐ ☐☐☐☐ ☐ ☐☐☐☐ ☐☐ ☐☐ ☐☐

Answer: A (LEAVE A REPLY)

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

NEW QUESTION: 130

account1□□□ □□□ Azure Cosmos DB for NoSQL □□□ □□□□.

```
account1 0000 0000 0000 000000 000 0000 000 container1000 00000 000 00000. container10 000 00 00 000 00 0000 000 0000000000 00 0000 0
```

□ □ □ □ □ □ . □ □ □ □ Vehicle-Identification-Number □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ .

□□□□1□ □□ □□□ □□□ □□□□ □□□. □□ □□□ □□ □□ □□□ □□□□ □□□.

- □□ □ □□ □□□ □□□□□□.

- □□□ □□ □ □□ □□□ □□□□□□.

□□□ □□□□ □□□?

A. □□ □□ □□(VIN) □□□ □□□□ □□□ □□ □□□□□.

B. ☐☐ ☐☐☐ ☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐ ☐☐☐☐.

C. ☐☐ ☐☐☐ ☐☐☐☐ ☐☐☐ ☐☐ ☐☐☐☐.

D. □□ □□□ □□ □□ □□(VIN) □□ □□ □□□□ □□□□ □□ □□ □□□□□□.

Answer: D (LEAVE A REPLY)

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: 131

☐ ☐ ☐ ☐ ☐

Azure Cosmos DB for NoSQL ☐☐☐ ☐☐☐☐☐ ☐☐☐☐.

□□□ □□□ □□□ □□□□ □□□ □□□□□ Azure Cosmos DB SDK□ □□□□ □□□.

□□□ □□□ □□□□ □□□? □□□ □□□□□ □□□□□ □□□ □□□ □□□□.

□□: □□ □□□ 1□□□□.

Answer Area

RequestOptions property to set:

AccessCondition

ConsistencyLevel

SessionToken

Document property that will be compared:

etag

id

rid

Answer:

Answer Area

RequestOptions property to set:

AccessCondition

ConsistencyLevel

SessionToken

Document property that will be compared:

_etag

_id

_rid

Explanation:

Box 1: ConsistencyLevel

The ItemRequestOptions Class ConsistencyLevel property gets or sets the consistency level required for the request in the Azure Cosmos DB service. Azure Cosmos DB offers 5 different consistency levels. Strong, Bounded Staleness, Session, Consistent Prefix and Eventual - in order of strongest to weakest consistency.

Box 2: _etag

The ItemRequestOptions class helped us implement optimistic concurrency by specifying that we wanted the SDK to use the If-Match header to allow the server to decide whether a resource should be updated. The If-Match value is the ETag value to be checked against. If the ETag value matches the server ETag value, the resource is updated.

Reference:

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.cosmos.itemrequestoptions>

<https://cosmosdb.github.io/labs/dotnet/labs/10-concurrency-control.html>

Azure Cosmos DB for NoSQL Container1 is a container for documents. Container1 is a container for documents. App1 is an application that uses Container1. App1 is an application that uses Container1.

- Container1 is a container for documents.
- App1 is an application that uses Container1.

App1 is an application that uses Container1. App1 is an application that uses Container1. App1 is an application that uses Container1.

- A. Container1 is a container for documents.
- B. Container1 is a container for documents.
- C. Container1 is a container for documents.
- D. Container1 is a container for documents.

Answer: B (LEAVE A REPLY)

The multiple document model is the optimal choice for your Azure Cosmos DB for NoSQL application. It keeps development and operational efforts as low as possible while cleanly satisfying the requirements. Storing each version as a separate document prevents the application from hitting the strict maximum limit of 2 MB per item in Azure Cosmos DB, which would eventually occur under a single-document approach as the version history grows.

Compare Multiple Document Models

Azure Cosmos DB for NoSQL functions best when documents are isolated, right-sized, and predictable.

Model Type	Development Effort	Operational Effort & Scale
Separate documents per entity version	Low. Standard NoSQL pattern. Write new versions as separate items with a shared entityId partition key.	Low. Efficient indexing, no risk of hitting the 2MB document limit, and cheap RU costs.
One document combining all entities per version	High. Complex application logic required to aggregate completely separate entities into global version state arrays.	High. Documents grow dangerously fast. Frequent cross-partition concurrency issues during updates.

Reference:

<https://learn.microsoft.com/en-us/samples/azure-samples/cosmos-db-design-patterns/document-versioning/>

NEW QUESTION: 133

DB1 is a database. Container1 is a container for documents. Azure Cosmos DB for NoSQL is a database. Azure Cosmos DB SDK is a library. App1 is an application that uses Container1. App1 is an application that uses Container1.

- A. Container1 is a container for documents.
- B. Container1 is a container for documents.
- C. DB1 is a database.
- D. Container1 is a container for documents.

Answer: (SHOW ANSWER)

NEW QUESTION: 134

Azure Cosmos DB for NoSQL is a database. Azure Cosmos DB for NoSQL is a database. Azure Cosmos DB for NoSQL is a database.

- Which Log Analytics workspace can you use to monitor the health of your Azure Cosmos DB database?
- A. Azure Log Analytics
 - B. Azure Monitor
 - C. Azure Log Analytics workspace
 - D. Azure Monitor workspace

Answer: C (LEAVE A REPLY)

NEW QUESTION: 135

Azure Cosmos DB for NoSQL uses a container-based architecture. A container is a logical unit of storage and retrieval. A container can be created with a specific configuration. Which of the following is a valid configuration for a container?

Container1.DefaultTimeToLive = 1000

Container1.DefaultTimeToLive = 1000000

Container1.DefaultTimeToLive = 1000000000

Container1.DefaultTimeToLive = 1000000000000

Answer: (SHOW ANSWER)

NEW QUESTION: 136

Which of the following is a valid configuration for a container?

Container1.DefaultTimeToLive = 1000

Container1.DefaultTimeToLive = 1000000

Container1.DefaultTimeToLive = 1000000000

Container1.DefaultTimeToLive = 1000000000000

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.

Modifv the settinas of DB1.

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.

DP-420-KR 00 0000 000000 00 DumpTop 00 00000 0000 DP-420-KR 00! DumpTop 0 00 DP-420-KR 00 0000 00000000, DumpTop DP-420-KR 00 0000 0000000000 0000 0000 00000. 00000 0000 00000 00 DumpTop DP-420-KR 0000 0000000. <https://www.dumptop.com/Microsoft/DP-420-KR-dump.html> (281 Q&As Dumps, 30%OFF Special Discount: KrDump)

DP-420-KR 00 0000 000000 00 DumpTop 00 00000 0000 DP-420-KR 00! DumpTop 0 00 DP-420-KR 00 0000 00000000, DumpTop DP-420-KR 00 0000 0000000000 0000 0000 00000. 00000 0000 00000 00 DumpTop DP-420-KR 0000 0000000. <https://www.dumptop.com/Microsoft/DP-420-KR-dump.html> (281 Q&As Dumps, 30%OFF Special Discount: KrDump)