

Microsoft.DP-420.vSep-2024.by.Anie.46q

Number: DP-420
Passing Score: 800
Time Limit: 120
File Version: 5.0

Exam Code: DP-420

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



Exam A

QUESTION 1

You have a database in an Azure Cosmos DB for NoSQL account. The database contains a container named container1. The indexing mode container1 is set to none. You configure Azure Cognitive Search to extract data from container1 and make the data searchable. You discover that the Cognitive Search index is missing all the data from the Azure Cosmos DB index. What should you do to resolve the issue?

- A. Modify The index attributes in Cognitive Search to searchable.
- B. Modify the index attributes in Cognitive Search to Retrievable.
- C. Change the indexing mode of container 1 to
- D. Modify the indexing policy of container 1 to exclude the / * path

Correct Answer: C

Section:

QUESTION 2

DRAG DROP

You have an Azure Synapse Analytics workspace named workspace1 that contains a server less SQL pool.

You have an Azure Table Storage account that stores operational data.

You need to replace the Table storage account with Azure Cosmos DB for NoSQL The solution must meet the following requirements:

- Support Queries from the server less SQL pool.
- Only pay for analytical compute when running queries.
- Ensure that analytical processes do

NOTE: affect operational processes.

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

Answer:

Actions

- In workspace1, create a dedicated SQL pool.
- In the Azure Cosmos DB account, create a table that has unlimited storage capacity.
- Create an Azure Cosmos DB for NoSQL account.
- Enable Azure Synapse Link.
- Create a database and a container that has Analytical store enabled.

Answer Area

- 1 Create an Azure Cosmos DB for NoSQL account.
- 2 Enable Azure Synapse Link.
- 3 Create a database and a container that has Analytical store enabled.

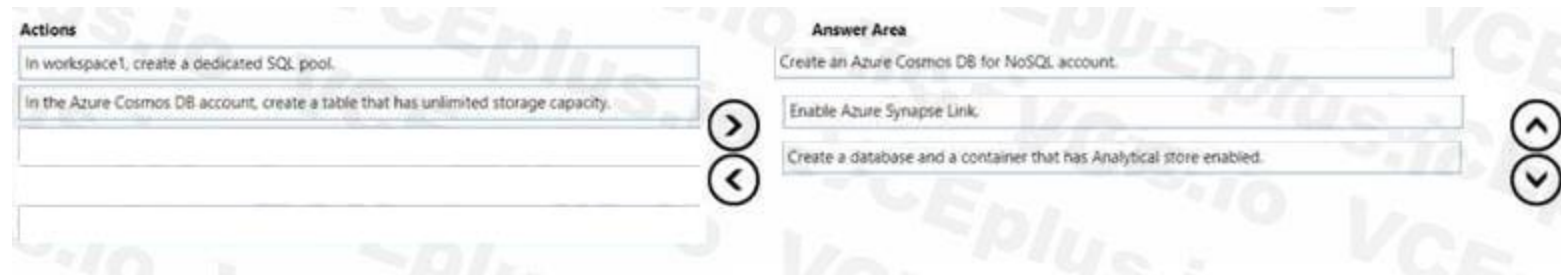
Select and Place:

Actions

- In workspace1, create a dedicated SQL pool.
- In the Azure Cosmos DB account, create a table that has unlimited storage capacity.
- Create an Azure Cosmos DB for NoSQL account.
- Enable Azure Synapse Link.
- Create a database and a container that has Analytical store enabled.

Answer Area

Correct Answer:



Section:

Explanation:

QUESTION 3

You have an Azure Cosmos DB Core (SQL) API account that is used by 10 web apps.

You need to analyze the data stored in the account by using Apache Spark to create machine learning models. The solution must NOT affect the performance of the web apps.

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

NOTE: Each correct selection is worth one point.

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

Correct Answer: A, D

Section:

Explanation:

Reference:

<https://github.com/microsoft/MCW-Cosmos-DB-Real-Time-Advanced-Analytics/blob/main/Handson%20lab/HOL%20step-by%20step%20-%20Cosmos%20DB%20realtime%20advanced%20analytics.md>

QUESTION 4

You have an Azure Cosmos DB for NoSQL account.

The change feed is enabled on a container named invoice.

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

What is received by the Azure function?

- A. all the properties of the updated items
- B. only the partition key and the changed properties of the updated items
- C. all the properties of the original items and the updated items
- D. only the changed properties and the system-defined properties of the updated items

Correct Answer: A

Section:

Explanation:

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

The Azure function that has a trigger on the change feed receives all the properties of the updated items². The change feed does not include the original items or only the changed properties. The change feed also includes some system- defined properties such as `_ts` (the last modified timestamp) and `_lsn` (the logical sequence number)³.

Therefore, the correct answer is:

- A. all the properties of the updated items

QUESTION 5

You have an Azure Cosmos DB database named databaset contains a container named container1. The container1 container store product data and has the following indexing policy.

```
{
  "indexingMode": "consistent",
  "includedPaths":
  [
    {
      "path": "/product/category/?"
    },
    {
      "path": "/product/brand/?"
    }
  ],
  "excludedPaths":
  [
    {
      "path": "/*"
    },
    {
      "path": "/product/brand"
    }
  ]
}
```

Which path will be indexed?

- A. /product/brand
- B. /product/category
- C. /product/[]/category
- D. /product/brand/tailspin

Correct Answer: A

Section:

Explanation:

The indexing policy has an includedPaths array that contains only one path: /product/brand/? . This means that only the properties under /product/brand will be indexed. The ? symbol indicates that only scalar values will be indexed, not arrays or objects1.

The excludedPaths array contains a single path: /* . This means that all other properties will be excluded from indexing. The * symbol indicates a wildcard that matches any property name1.

Therefore, the paths /product/category , /product/[]/category , and /product/brand/tailspin will not be indexed.

QUESTION 6

HOTSPOT

You have a database named db1 in an Azure Cosmos DB for NoSQL account named account1. The db1 database has a manual throughput of 4,000 request units per second (RU/s).

You need to move db1 from manual throughput to autoscale throughput by using the Azure CLI. The solution must provide a minimum of 4,000 RU/s and a maximum of 40,000 RU/s.

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



NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

```
az cosmosdb sql database throughput
```

-a "account1" \
-g "cosmosdbrg" \
-n "db1" \
--throughput

migrate
show
update

400
4000
40000

Answer Area:

Answer Area

```
az cosmosdb sql database throughput
```

-a "account1" \
-g "cosmosdbrg" \
-n "db1" \
--throughput

migrate
show
update

400
4000
40000

Section:

Explanation:

Migrate

40000

According to the Azure CLI reference¹, you need to use the `az cosmosdb sql database throughput migrate` command to migrate the throughput of the SQL database between autoscale and manually provisioned. You also need to use the `--throughput-type` parameter to specify the type of throughput to migrate to, and the `--max-throughput` parameter to specify the maximum throughput resource can scale to (RU/s).

To complete the CLI statements, you should replace the missing values with:

`--throughput-type autoscale`

`--max-throughput 40000`

The final command should look like this:

```
az cosmosdb sql database throughput migrate \  
--account-name account1 \  
--name db1 \  
--resource-group rg1 \  
--throughput-type autoscale \  
--max-throughput 40000
```

QUESTION 7

HOTSPOT

You have a container named container1 in an Azure Cosmos DB for NoSQL account named account1.

You configure container1 to use Always Encrypted by using an encryption policy as shown in the C# and the Java exhibits. (Click the C# tab to view the encryption policy in C#. Click the Java tab to see the encryption policy in Java.)

```
var path1 = new ClientEncryptionIncludedPath
{
    Path = "/creditcard",
    ClientEncryptionKeyId = "encryptionkey",
    EncryptionType = EncryptionType.Randomized.ToString(),
    EncryptionAlgorithm = DataEncryptionKeyAlgorithm.AEAD_AES_256_CBC_HMAC_SHA256.ToString()
};
var path2 = new ClientEncryptionIncludedPath
{
    Path = "/SSN",
    ClientEncryptionKeyId = "encryptionkey",
    EncryptionType = EncryptionType.Deterministic.ToString(),
    EncryptionAlgorithm = DataEncryptionKeyAlgorithm.AEAD_AES_256_CBC_HMAC_SHA256.ToString()
};

await database.DefineContainer("container1", "/partitionkey")
    .WithClientEncryptionPolicy()
    .WithIncludedPath(path1)
    .WithIncludedPath(path2)
    .Attach()
    .CreateAsync();

ClientEncryptionIncludedPath path1 = new ClientEncryptionIncludedPath();
path1.path = "/creditcard";
path1.clientEncryptionKeyId = "encryptionkey";
path1.encryptionType = CosmosEncryptionType.RANDOMIZED;
path1.encryptionAlgorithm = CosmosEncryptionAlgorithm.AEAES_256_CBC_HMAC_SHA_256;

ClientEncryptionIncludedPath path2 = new ClientEncryptionIncludedPath();
path2.path = "/SSN";
path2.clientEncryptionKeyId = "encryptionkey";
path2.encryptionType = CosmosEncryptionType.DETERMINISTIC;
path2.encryptionAlgorithm = CosmosEncryptionAlgorithm.AEAES_256_CBC_HMAC_SHA_256;

ClientEncryptionIncludedPath[] paths = new ArrayList<>();
paths.add(path1);
paths.add(path2);

CosmosContainerProperties containerProperties =
    new CosmosContainerProperties("container1", "/partitionkey");
containerProperties.setClientEncryptionPolicy(new ClientEncryptionPolicy(paths));
database.createEncryptionContainerAsync(containerProperties);
```



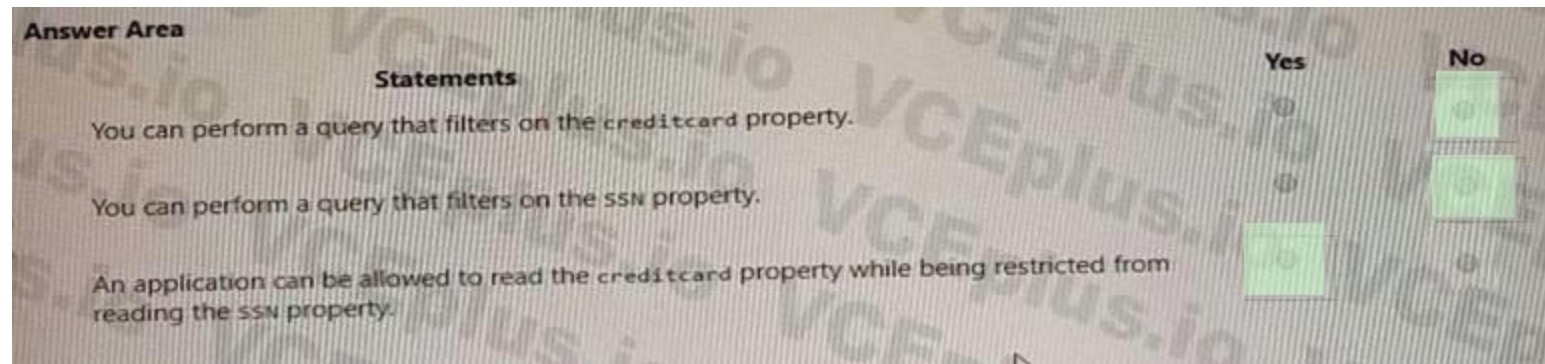
For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Hot Area:

| Statements | Yes | No |
|---|-----------------------|-----------------------|
| You can perform a query that filters on the creditcard property. | <input type="radio"/> | <input type="radio"/> |
| You can perform a query that filters on the ssn property. | <input type="radio"/> | <input type="radio"/> |
| An application can be allowed to read the creditcard property while being restricted from reading the ssn property. | <input type="radio"/> | <input type="radio"/> |

Answer Area:



Section:

Explanation:

According to the Azure Cosmos DB documentation¹, Always Encrypted is a feature designed to protect sensitive data, such as credit card numbers or national identification numbers, stored in Azure Cosmos DB. Always Encrypted allows clients to encrypt sensitive data inside client applications and never reveal the encryption keys to the database.

To use Always Encrypted, you need to define an encryption policy for each container that specifies which properties should be encrypted and which data encryption keys (DEK) should be used. The DEKs are stored in Azure Cosmos DB and are wrapped by customer-managed keys (CMK) that are stored in Azure Key Vault.

Based on the encryption policy shown in the exhibits, the creditcard property is encrypted with a DEK named dek1, and the SSN property is encrypted with a DEK named dek2. Both DEKs are wrapped by a CMK named cmk1.

To answer your statements:

You can perform a query that filters on the creditcard property = No. This is because the creditcard property is encrypted and cannot be used for filtering or sorting operations¹.

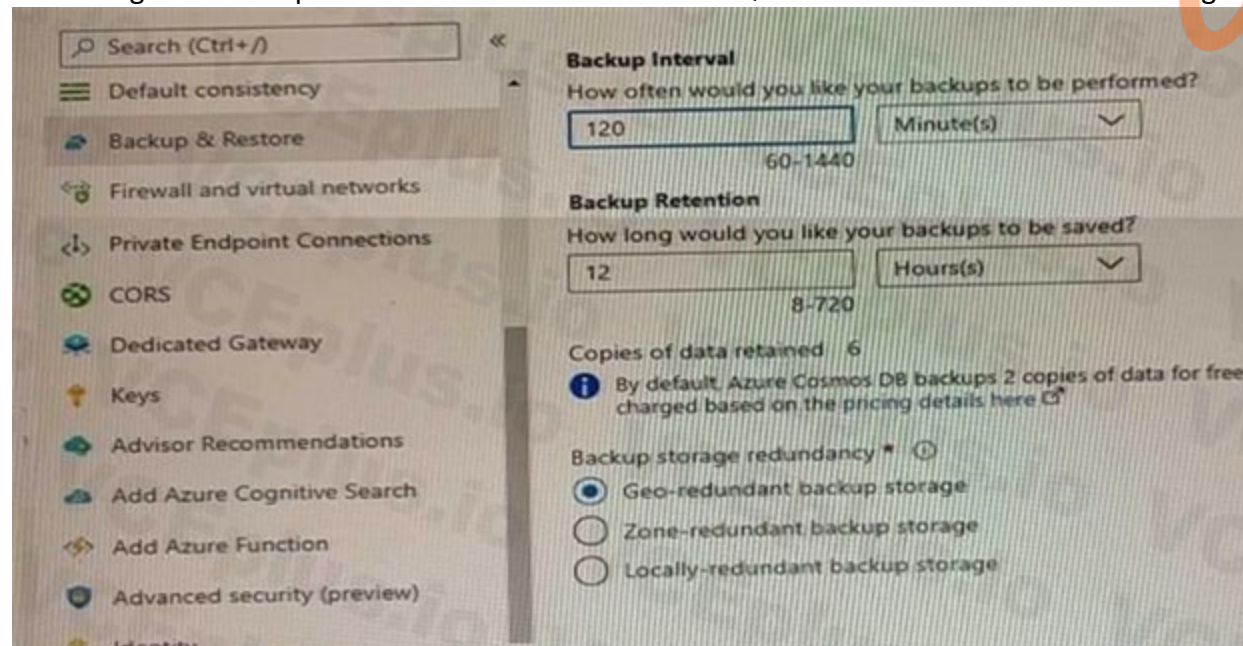
You can perform a query that filters on the SSN property = No. This is also because the SSN property is encrypted and cannot be used for filtering or sorting operations¹.

An application can be allowed to read the creditcard property while being restricted from reading the SSN property = Yes. This is possible by using different CMKs to wrap different DEKs and applying access policies on the CMKs in Azure Key Vault. For example, if you use cmk2 to wrap dek2 instead of cmk1, you can grant an application access to cmk1 but not cmk2, which means it can read the creditcard property but not the SSN property².

QUESTION 8

HOTSPOT

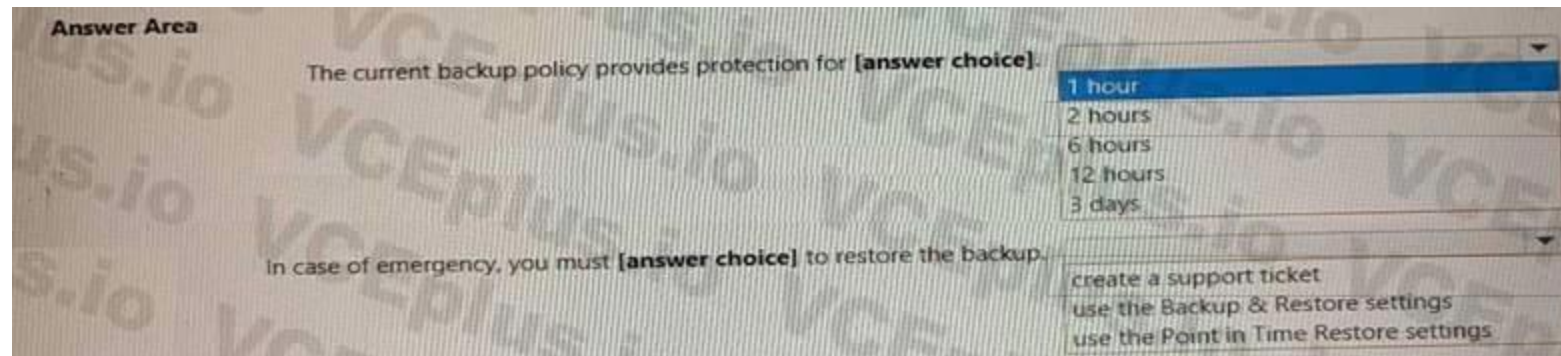
You configure a backup for an Azure Cosmos DB for NoSQL account as shown in the following exhibit.



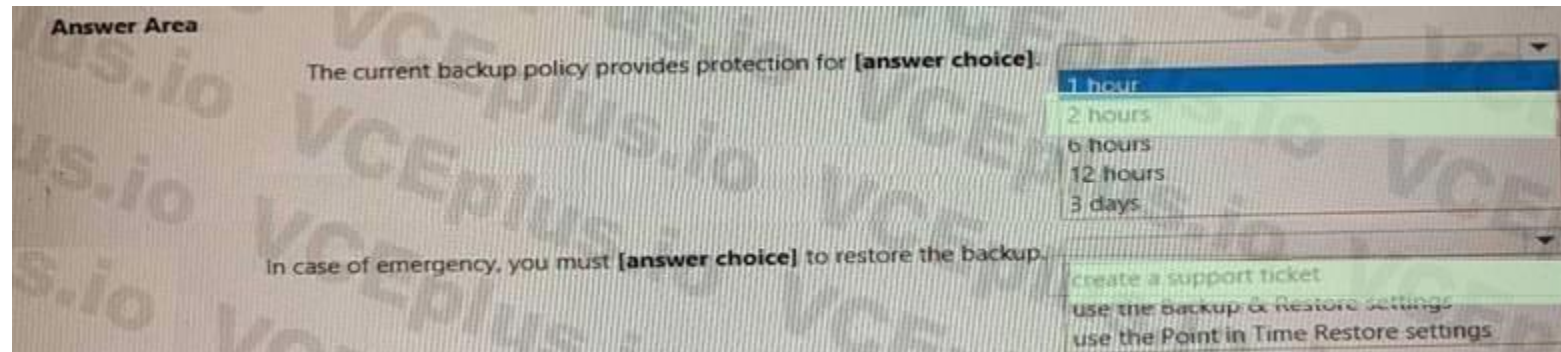
Use the drop-down menus to select the answer choice that completes each statement based on the information presented in the graphic.

NOTE: Each correct selection is worth one point.

Hot Area:



Answer Area:



Section:

Explanation:

Box 1 = The current backup policy provides protection for: 2 Hours

Azure Cosmos DB automatically takes backups of your data at regular intervals. The backup interval and the retention period can be configured from the Azure portal. You can also choose between two backup modes: periodic backup mode and continuous backup mode. Periodic backup mode is the default mode for all existing accounts and it takes a full backup of your database every 4 hours by default. Continuous backup mode is a new mode that allows you to restore to any point of time within either 7 or 30 days¹.

For your scenario, based on the exhibit, you have configured a backup for an Azure Cosmos DB for NoSQL account using the periodic backup mode with a backup interval of 1 hour and a retention period of 2 hours. This means that Azure Cosmos DB will take a full backup of your database every hour and keep only the latest two backups. Therefore, the current backup policy provides protection for 2 hours.

Box 2: In case of emergency, you must (answer choice) to restore the backup = create a support ticket Azure Cosmos DB automatically takes backups of your data at regular intervals. You can configure the backup interval and the retention period from the Azure portal. You can also choose between two backup modes: periodic backup mode and continuous backup mode. Periodic backup mode is the default mode for all existing accounts and it takes a full backup of your database every 4 hours by default. Continuous backup mode is a new mode that allows you to restore to any point of time within either 7 or 30 days¹.

For your scenario, based on the exhibit, you have configured a backup for an Azure Cosmos DB for NoSQL account using the periodic backup mode with a backup interval of 1 hour and a retention period of 2 hours. This means that Azure Cosmos DB will take a full backup of your database every hour and keep only the latest two backups. In case of emergency, you must create a support ticket to restore the backup. This is the answer to your question.

To restore data from a periodic backup, you need to create a support request with Azure Cosmos DB team and provide the following information:

The name of your Azure Cosmos DB account

The name of the database or container that you want to restore

The date and time (in UTC) that you want to restore from

The name of the target Azure Cosmos DB account where you want to restore the data The name of the target resource group where you want to restore the data The Azure Cosmos DB team will then initiate the restore process and notify you when it is completed².

QUESTION 9

HOTSPOT

You have an Azure Cosmos DB for NoSQL account that frequently receives the same three queries.

You need to configure indexing to minimize RUs consumed by the queries.

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

NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

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

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

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

Composite
Range
Spatial

Composite
Range
Spatial

Composite
Range
Spatial

Answer Area:

Answer Area

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

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

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

Composite
Range
Spatial

Composite
Range
Spatial

Composite
Range
Spatial

Section:

Explanation:

Box 1 = Range

Azure Cosmos DB supports three types of indexes: range, spatial and composite. For the query you provided, which is an equality query on a single property, the best type of index to use is range. Range index is based on an ordered tree-like structure and it is used for equality queries, range queries and checking for the presence of a property¹. Range index also supports any string or number².

Box 2 = Composite

Azure Cosmos DB supports three types of indexes: range, spatial and composite. For the query you provided, which is an order by query on two properties, the best type of index to use is composite. Composite index is used for optimizing order by queries on multiple properties¹. Composite index allows you to specify a list of property paths and sort orders that are used for ordering items².

Box 3 = spatial

Azure Cosmos DB supports three types of indexes: range, spatial and composite. For the query you provided, which is a spatial query on a point property, the best type of index to use is spatial. Spatial index is used for querying items based on their location or proximity to a given point¹. Spatial index supports point, polygon and linestring data types².

QUESTION 10

HOTSPOT

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

```
{
  "lastName": "Cartwright",
  "parents": [
    {
      "firstName": "Elvira",
      "role": "mother",
      "age": 64
    },
    {
      "firstName": "Randolph",
      "role": "father",
      "age": 67
    }
  ],
  "children": [
    {
      "grade": 5,
      "name": "Pat",
      "age": 13,
      "gender": "male"
    }
  ]
}
```

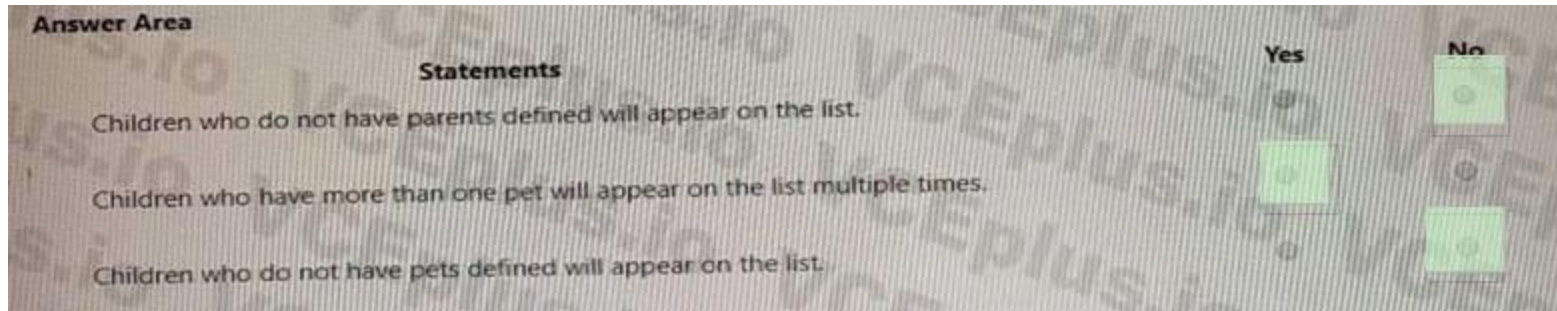


For each of the following statements, select Yes if the statement is true. otherwise, select No.
NOTE: Each correct selection is worth one point.

Hot Area:

| Statements | Yes | No |
|---|--------------------------|--------------------------|
| Children who do not have parents defined will appear on the list. | <input type="checkbox"/> | <input type="checkbox"/> |
| Children who have more than one pet will appear on the list multiple times. | <input type="checkbox"/> | <input type="checkbox"/> |
| Children who do not have pets defined will appear on the list. | <input type="checkbox"/> | <input type="checkbox"/> |

Answer Area:



Section:

Explanation:

Children who do not have parents defined will appear on the list = NO

Children who do not have parents defined will not appear on the list. This is because the document schema defines the children property as an array of objects that contain the firstName and gender properties of each child, as well as a parents property that references the id values of the parents. If a child does not have parents defined, it means that the parents property is either missing or empty for that child. Therefore, such a child will not be included in the list of children who have parents defined.

Children who have more than one pet will appear on the list multiple times. = Yes

Children who have more than one pet will appear on the list multiple times. This is because the document schema defines the pets property as an array of objects that contain the givenName and type properties of each pet, as well as a children property that references the id values of the children who own the pet. If a child has more than one pet, it means that the child's id value will appear in the children property of multiple pet objects. Therefore, such a child will be included in the list of children who have pets multiple times.

Children who do not have pets defined will appear on the list = No

Children who do not have pets defined will not appear on the list. This is because the document schema defines the pets property as an array of objects that contain the givenName and type properties of each pet, as well as a children property that references the id values of the children who own the pet. If a child does not have pets defined, it means that the child's id value does not appear in the children property of any pet object. Therefore, such a child will not be included in the list of children who have pets defined.

QUESTION 11

HOTSPOT

You have the Azure Cosmos DB for NoSQL containers shown in the following table.



| Name | DefaultTimeToLive |
|------------|-------------------|
| container1 | -1 |
| container2 | null |
| container3 | 60 |

You have the items shown in the following table.

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

When will each item expire? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

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

Answer Area:

Answer Area

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

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

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

Section:

Explanation:

QUESTION 12

HOTSPOT

You have an Azure Cosmos DB for NoSQL account named account1.

You plan to implement the integrated cache for account1.

You need to configure the connectivity mode and the consistency level for requests that target account1. The solution must maximize consistency while using the integrated cache.

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

NOTE: Each correct selection is worth one point.

Hot Area:

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

Answer Area:

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

Section:

Explanation:

QUESTION 13

HOTSPOT

You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account named accounts

You configure account1 to send resource logs to Log Analytics.

You need to identify any containers in account1 that were deleted during the past 24 hours.

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

NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

CDBControlPlaneRequests
CDBDataPlaneRequests
CDBQueryRuntimeStatistics

| where TimeGenerated >= now(-1d)

| where == "SqlContainersDelete"

OperationName
OperationType
Result

Answer Area:



Answer Area

| | |
|---------------------------|---|
| CDBControlPlaneRequests | ▼ |
| CDBControlPlaneRequests | |
| CDBDataPlaneRequests | |
| CDBQueryRuntimeStatistics | |

| where TimeGenerated >= now(-1d)

| where

| | | |
|---------------|---|--------------------------|
| OperationName | ▼ | == "SqlContainersDelete" |
| OperationName | | |
| OperationType | | |
| Result | | |

Section:

Explanation:

QUESTION 14

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

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have an Azure Cosmos DB Core (SQL) API account named account 1 that uses autoscale throughput.

You need to run an Azure function when the normalized request units per second for a container in account1 exceeds a specific value.

Solution: You configure an Azure Monitor alert to trigger the function.

Does this meet the goal?

- A. Yes
- B. No



Correct Answer: A

Section:

Explanation:

You can set up alerts from the Azure Cosmos DB pane or the Azure Monitor service in the Azure portal.

Note: Alerts are used to set up recurring tests to monitor the availability and responsiveness of your Azure Cosmos DB resources. Alerts can send you a notification in the form of an email, or execute an Azure Function when one of your metrics reaches the threshold or if a specific event is logged in the activity log.

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

QUESTION 15

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

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have an Azure Cosmos DB Core (SQL) API account named account 1 that uses autoscale throughput.

You need to run an Azure function when the normalized request units per second for a container in account1 exceeds a specific value.

Solution: You configure the function to have an Azure CosmosDB trigger.

Does this meet the goal?

- A. Yes
- B. No

Correct Answer: B

Section:

Explanation:

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>

QUESTION 16

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

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have an Azure Cosmos DB Core (SQL) API account named account 1 that uses autoscale throughput.

You need to run an Azure function when the normalized request units per second for a container in account1 exceeds a specific value.

Solution: You configure an application to use the change feed processor to read the change feed and you configure the application to trigger the function.

Does this meet the goal?

- A. Yes
- B. No

Correct Answer: B

Section:**Explanation:**

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>

QUESTION 17

You have a container in an Azure Cosmos DB Core (SQL) API account. The container stores telemetry data from IoT devices. The container uses telemetryId as the partition key and has a throughput of 1,000 request units per second (RU/ s). Approximately 5,000 IoT devices submit data every five minutes by using the same telemetryId value.

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

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

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

You need to reduce the amount of request units (RUs) consumed by the analytics application.

What should you do?

- A. Decrease the offerThroughput value for the container.
- B. Increase the offerThroughput value for the container.
- C. Move the data to a new container that has a partition key of deviceId.
- D. Move the data to a new container that uses a partition key of date.

Correct Answer: C

Section:**Explanation:**

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

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

QUESTION 18

The settings for a container in an Azure Cosmos DB Core (SQL) API account are configured as shown in the following exhibit.

The screenshot shows the 'Settings' tab for a container in Azure Cosmos DB. The 'Time to Live' section has three radio buttons: 'Off', 'On (no default)' (which is selected), and 'On'. The 'Geospatial Configuration' section has two radio buttons: 'Geography' (which is selected) and 'Geometry'. Below these sections, the 'Partition key' is set to '/productName'.



Which statement describes the configuration of the container?

- A. All items will be deleted after one year.
- B. Items stored in the collection will be retained always, regardless of the items time to live value.

- C. Items stored in the collection will expire only if the item has a time to live value.
- D. All items will be deleted after one hour.

Correct Answer: C

Section:

Explanation:

When DefaultTimeToLive is -1 then your Time to Live setting is On (No default) Time to Live on a container, if present and the value is set to "-1", it is equal to infinity, and items don't expire by default.

Time to Live on an item:

This Property is applicable only if DefaultTimeToLive is present and it is not set to null for the parent container.

If present, it overrides the DefaultTimeToLive value of the parent container.

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

QUESTION 19

You have an Azure Cosmos DB Core (SQL) API account that uses a custom conflict resolution policy.

The account has a registered merge procedure that throws a runtime exception.

The runtime exception prevents conflicts from being resolved.

You need to use an Azure function to resolve the conflicts.

What should you use?

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

Correct Answer: D

Section:

Explanation:

The Azure Cosmos DB Trigger uses the Azure Cosmos DB Change Feed to listen for inserts and updates across partitions. The change feed publishes inserts and updates, not deletions.

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

QUESTION 20

The following is a sample of a document in orders.

```
{
  "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"
}
```

The orders container uses customerId as the partition key.

You need to provide a report of the total items ordered per month by item type. The solution must meet the following requirements:
Ensure that the report can run as quickly as possible.
Minimize the consumption of request units (RUs).
What should you do?

- A. Configure the report to query orders by using a SQL query.
- B. Configure the report to query a new aggregate container. Populate the aggregates by using the change feed.
- C. Configure the report to query orders by using a SQL query through a dedicated gateway.
- D. Configure the report to query a new aggregate container. Populate the aggregates by using SQL queries that run daily.

Correct Answer: B

Section:

Explanation:

QUESTION 21

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

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account.

You need to make the contents of container1 available as reference data for an Azure Stream Analytics job.

Solution: You create an Azure Synapse pipeline that uses Azure Cosmos DB Core (SQL) API as the input and Azure Blob Storage as the output.

Does this meet the goal?

- A. Yes
- B. No



Correct Answer: B

Section:

Explanation:

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>

QUESTION 22

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

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account.

You need to make the contents of container1 available as reference data for an Azure Stream Analytics job.

Solution: You create an Azure Data Factory pipeline that uses Azure Cosmos DB Core (SQL) API as the input and Azure Blob Storage as the output.

Does this meet the goal?

- A. Yes

B. No

Correct Answer: B

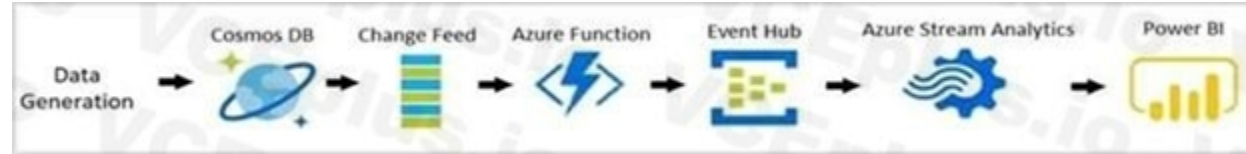
Section:

Explanation:

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>

QUESTION 23

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

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account.

You need to make the contents of container1 available as reference data for an Azure Stream Analytics job.

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

Does this meet the goal?

A. Yes

B. No



Correct Answer: A

Section:

Explanation:

QUESTION 24

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account. Upserts of items in container1 occur every three seconds.

You have an Azure Functions app named function1 that is supposed to run whenever items are inserted or replaced in container1.

You discover that function1 runs, but not on every upsert.

You need to ensure that function1 processes each upsert within one second of the upsert.

Which property should you change in the Function.json file of function1?

A. checkpointInterval

B. leaseCollectionsThroughput

C. maxItemsPerInvocation

D. feedPollDelay

Correct Answer: D

Section:

Explanation:

With an upsert operation we can either insert or update an existing record at the same time.

FeedPollDelay: The time (in milliseconds) for the delay between polling a partition for new changes on the feed, after all current changes are drained. Default is 5,000 milliseconds, or 5 seconds.

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-cosmosdbv2-trigger>

QUESTION 25

You have the following query.

```
SELECT * FROM ?
```

```
WHERE c.sensor = "TEMP1"
```

```
AND c.value < 22
```

```
AND c.timestamp >= 1619146031231
```

You need to recommend a composite index strategy that will minimize the request units (RUs) consumed by the query.

What should you recommend?

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

Correct Answer: A

Section:

Explanation:

If a query has a filter with two or more properties, adding a composite index will improve performance.

Consider the following query:

```
SELECT * FROM c WHERE c.name = "Tim" and c.age > 18
```

In the absence of a composite index on (name ASC, and age ASC), we will utilize a range index for this query. We can improve the efficiency of this query by creating a composite index for name and age.

Queries with multiple equality filters and a maximum of one range filter (such as >, <, <=, >=, !=) will utilize the composite index.

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

QUESTION 26

You plan to create an Azure Cosmos DB Core (SQL) API account that will use customer-managed keys stored in Azure Key Vault.

You need to configure an access policy in Key Vault to allow Azure Cosmos DB access to the keys.

Which three permissions should you enable in the access policy? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Wrap Key
- B. Get
- C. List
- D. Update
- E. Sign
- F. Verify
- G. Unwrap Key

Correct Answer: A, B, G

Section:

Explanation:

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

QUESTION 27

You need to configure an Apache Kafka instance to ingest data from an Azure Cosmos DB Core (SQL) API account. The data from a container named telemetry must be added to a Kafka topic named iot.

The solution must store the data in a compact binary format.

Which three configuration items should you include in the solution? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. "connector.class": "com.azure.cosmos.kafka.connect.source.CosmosDBSourceConnector"
- B. "key.converter": "org.apache.kafka.connect.json.JsonConverter"
- C. "key.converter": "io.confluent.connect.avro.AvroConverter"
- D. "connect.cosmos.containers.topicmap": "iot#telemetry"
- E. "connect.cosmos.containers.topicmap": "iot"
- F. "connector.class": "com.azure.cosmos.kafka.connect.source.CosmosDBSinkConnector"

Correct Answer: C, D, F

Section:**Explanation:**

C: Avro is binary format, while JSON is text.

F: Kafka Connect for Azure Cosmos DB is a connector to read from and write data to Azure Cosmos DB. The Azure Cosmos DB sink connector allows you to export data from Apache Kafka topics to an Azure Cosmos DB database. The connector polls data from Kafka to write to containers in the database based on the topics subscription.

D: Create the Azure Cosmos DB sink connector in Kafka Connect. The following JSON body defines config for the sink connector.

Extract:

```
"connector.class": "com.azure.cosmos.kafka.connect.sink.CosmosDBSinkConnector", "key.converter": "org.apache.kafka.connect.json.AvroConverter" "connect.cosmos.containers.topicmap": "hotels#kafka"
```

Incorrect Answers:

B: JSON is plain text.

Note, full example:

```
{
  "name": "cosmosdb-sink-connector",
  "config": {
    "connector.class": "com.azure.cosmos.kafka.connect.sink.CosmosDBSinkConnector",
    "tasks.max": "1",
    "topics": [
      "hotels"
    ],
    "value.converter": "org.apache.kafka.connect.json.AvroConverter",
    "value.converter.schemas.enable": "false",
    "key.converter": "org.apache.kafka.connect.json.AvroConverter",
    "key.converter.schemas.enable": "false",
    "connect.cosmos.connection.endpoint": "https://.
documents.azure.com:443/",
    "connect.cosmos.master.key": "",
    "connect.cosmos.databasename": "kafkaconnect",
    "connect.cosmos.containers.topicmap": "hotels#kafka"
  }
}
```

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/kafka-connector-sink>

<https://www.confluent.io/blog/kafka-connect-deep-dive-converters-serialization-explained/>

QUESTION 28

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

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

Correct Answer: C

Section:

Explanation:

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>



QUESTION 29

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account.

You need to provide a user named User1 with the ability to insert items into container1 by using rolebased access control (RBAC). The solution must use the principle of least privilege. Which roles should you assign to User1?

- A. CosmosDB Operator only
- B. DocumentDB Account Contributor and Cosmos DB Built-in Data Contributor
- C. DocumentDB Account Contributor only
- D. Cosmos DB Built-in Data Contributor only

Correct Answer: A

Section:

Explanation:

Cosmos DB Operator: Can provision Azure Cosmos accounts, databases, and containers. Cannot access any data or use Data Explorer.

Incorrect Answers:

B: DocumentDB Account Contributor can manage Azure Cosmos DB accounts. Azure Cosmos DB is formerly known as DocumentDB.

C: DocumentDB Account Contributor: Can manage Azure Cosmos DB accounts.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/role-based-access-control>

QUESTION 30

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

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

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

You write the following query.

```
AzureDiagnostics
```

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

What should you include in the query?

- A. | where OperationName startswith "AccountUpdateStart"
- B. | where OperationName startswith "SqlContainersDelete"
- C. | where OperationName startswith "MongoCollectionsThroughputUpdate"
- D. | where OperationName startswith "SqlContainersThroughputUpdate"

Correct Answer: A

Section:

Explanation:

The following are the operation names in diagnostic logs for different operations:

RegionAddStart, RegionAddComplete

RegionRemoveStart, RegionRemoveComplete

AccountDeleteStart, AccountDeleteComplete

RegionFailoverStart, RegionFailoverComplete

AccountCreateStart, AccountCreateComplete

AccountUpdateStart, AccountUpdateComplete

VirtualNetworkDeleteStart, VirtualNetworkDeleteComplete

DiagnosticLogUpdateStart, DiagnosticLogUpdateComplete

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

QUESTION 31

You have a database in an Azure Cosmos DB Core (SQL) API account. The database is backed up every two hours.

You need to implement a solution that supports point-in-time restore.

What should you do first?

- A. Enable Continuous Backup for the account.
- B. Configure the Backup & Restore settings for the account.
- C. Create a new account that has a periodic backup policy.
- D. Configure the Point In Time Restore settings for the account.

Correct Answer: A

Section:

Explanation:

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/provision-account-continuousbackup>

QUESTION 32

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

You need to create an Azure function that will access the database to retrieve records based on a variable named accountnumber. The solution must protect against SQL injection attacks.

How should you define the command statement in the function?

- A. cmd = "SELECT * FROM Persons p
WHERE p.accountnumber = 'accountnumber'"
- B. cmd = "SELECT * FROM Persons p
WHERE p.accountnumber = LIKE @accountnumber"

- C. cmd = "SELECT * FROM Persons p
WHERE p.accountnumber = @accountnumber"
- D. cmd = "SELECT * FROM Persons p
WHERE p.accountnumber = '' + accountnumber + ''"

Correct Answer: C

Section:

Explanation:

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>

QUESTION 33

HOTSPOT

You are developing an application that will connect to an Azure Cosmos DB for NoSQL account. The account has a single read region and one agonal read region. The regions are configured for automatic failover. The account has the following connect strings. (Line numbers are included for reference only.)

```

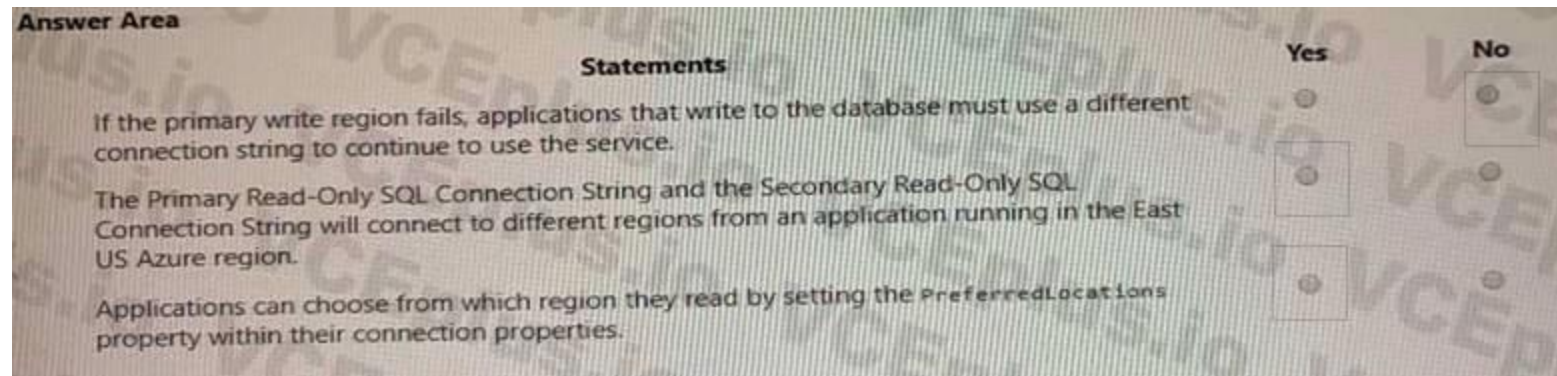
01 {
02   "connectionStrings": [
03     {
04       "connectionString":
05       "AccountEndpoint=https://contosodbaccount.documents.azure.com:443/;
06       AccountKey=MwUgRnGti4vErT2rFPFFdTFfYI9KyI9Kbe1RPGv70QdHo6VZ2i45TcJzrd4J80zYxrEATzyZh0M1nJaNFA==";
07       "description": "Primary SQL Connection String"
08     },
09     {
10       "connectionString":
11       "AccountEndpoint=https://contosodbaccount.documents.azure.com:443/;
12       AccountKey=gfThRnGti4vErT2rFPFFdTFfYI43529Kbe1RPGv70QdHo6VZ2i45TcJzrd4J80zYxrEATzyZh0M1nJaNFA==";
13       "description": "Secondary SQL Connection String"
14     },
15     {
16       "connectionString":
17       "AccountEndpoint=https://contosodbaccount.documents.azure.com:443/;
18       AccountKey=W6ykBclPHJoos6MdErT2rFPFFx9yI9Kbe1RPGv70QdHo6VZ2i45TcJzrd4J80zYxrEATzyZh0M1nJaNFA==";
19       "description": "Primary Read-Only SQL Connection String"
20     },
21     {
22       "connectionString":
23       "AccountEndpoint=https://contosodbaccount.documents.azure.com:443/;
24       AccountKey=k2DI0oY4Jc7QeUJqVGH3csda6EyI9Kbe1RPGv70QdHo6VZ2i45TcJzrd4J80zYxrEATzyZh0M1nJaNFA==";
25       "description": "Secondary Read-Only SQL Connection String"

```

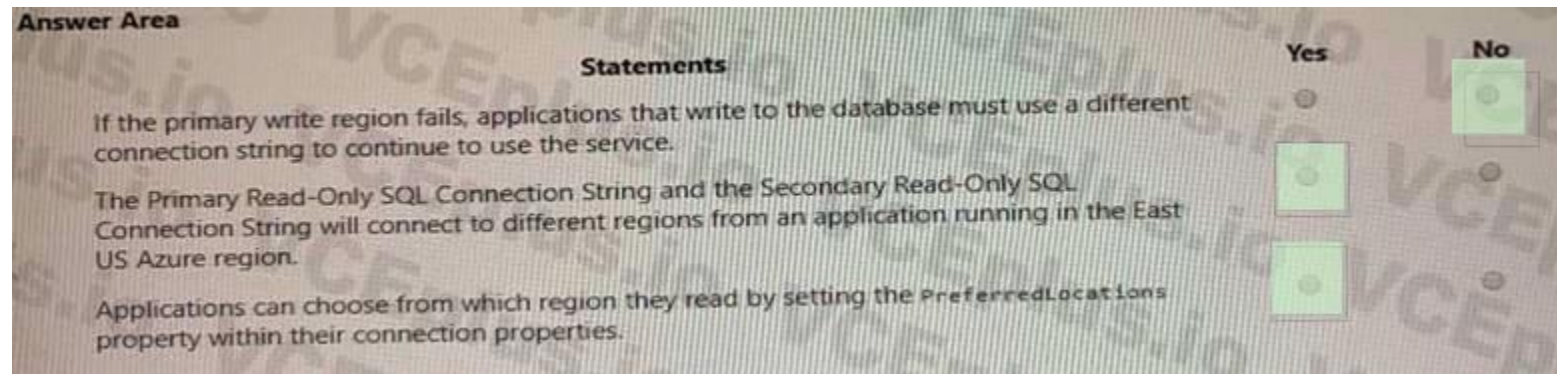
For each of the following statements, select Yes if the statement is true. otherwise, select No.

NOTE: Each correct selection is worth one point.

Hot Area:



Answer Area:



Section:

Explanation:

If the primary write region fails, applications that write to the database must use a different connection string to continue to use the service. = NO

You do not need to use a different connection string to continue to use the service if the primary write region fails. This is because Azure Cosmos DB supports automatic failover, which means that it will automatically switch the primary write region to another region in case of a regional outage². The application does not need to change the connection string or specify the failover priority³. The connection string contains a list of all the regions associated with your account, and

Azure Cosmos DB will route the requests to the appropriate region based on the availability and latency¹.

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 = Yes

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. This is because the primary read-only SQL connection string contains the endpoint for the East US region, which is the same as the primary write region. The secondary read-only SQL connection string contains the endpoint for the West US region, which is the additional read region.

Therefore, if an application running in the East US Azure region uses these connection strings, it will connect to different regions depending on which one it chooses.

Applications can choose from which region by setting the PreferredLocations property within their connection properties = Yes

Applications can choose from which region by setting the PreferredLocations property within their connection properties. This property allows you to specify a list of regions that you prefer to read from based on their proximity to your application². Azure Cosmos DB will route the requests to the appropriate region based on the availability and latency¹. You can also set the ApplicationRegion property to the region where your application is deployed, and Azure Cosmos

DB will automatically populate the PreferredLocations property based on the geo-proximity from that location¹.

QUESTION 34

HOTSPOT

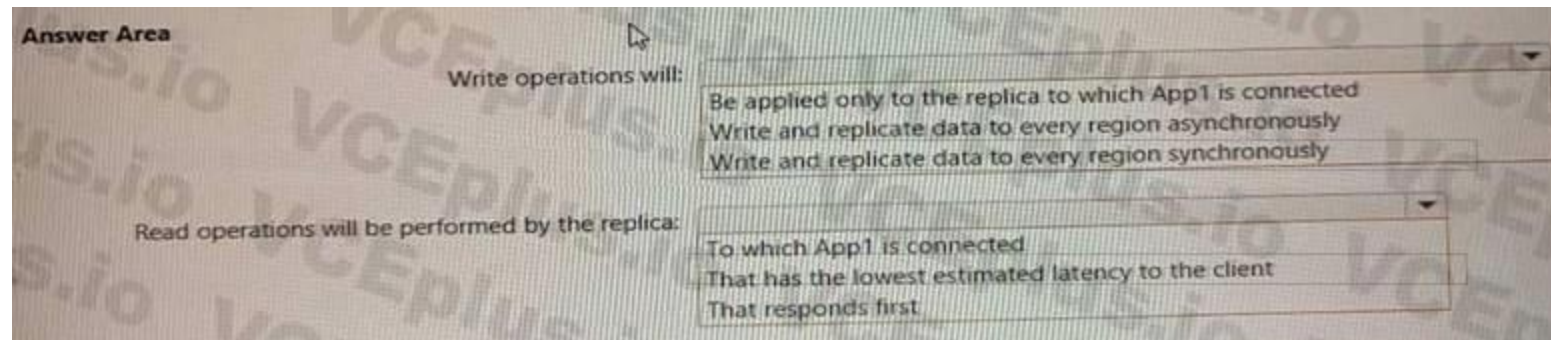
You have a multi-region Azure Cosmos DB account named account1 that has a default consistency level of strong.

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

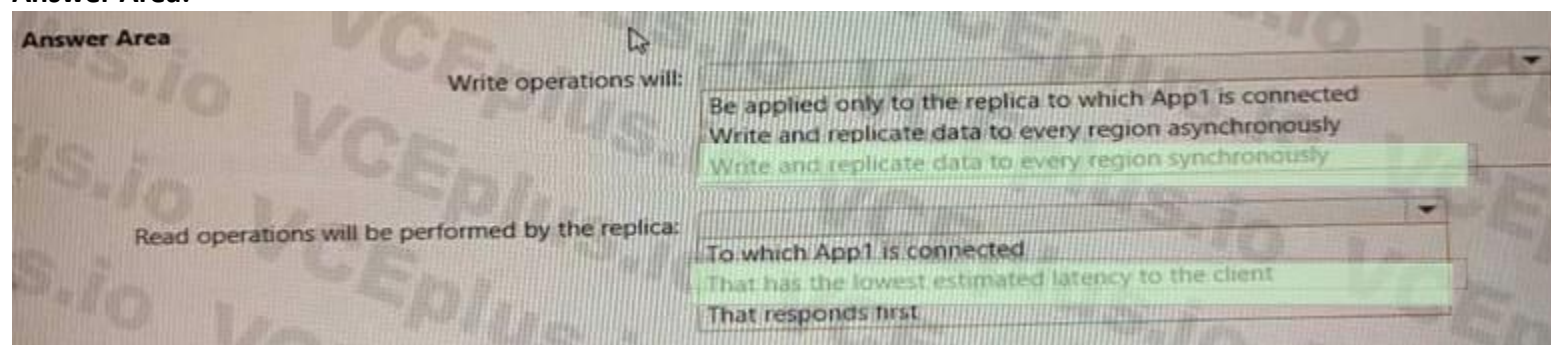
How will the read and write operations of App1 be handled? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:



Answer Area:



Section:

Explanation:

Box 1 = Write and replicate data to every region synchronously

This is because the write concern is mapped to the default consistency level configured on your Azure Cosmos DB account2, which is strong in this case. Strong consistency ensures that every write operation is synchronously committed to every region associated with your Azure Cosmos DB account1. The request level consistency level of session only applies to the read operations of App1.

Box 2: That has the lowest estimated latency to the client

This is because the read operations of App1 will use the session consistency level that is specified in the request options. Session consistency is a client-centric consistency model that guarantees monotonic reads, monotonic writes, and read-your-own-writes within a session. A session is scoped to a client connection or a stored procedure execution. Session consistency allows clients to read from any region that has the lowest latency to the client.

QUESTION 35

HOTSPOT

You have an Azure Cosmos DB account named account1 that has a default consistency level of session.

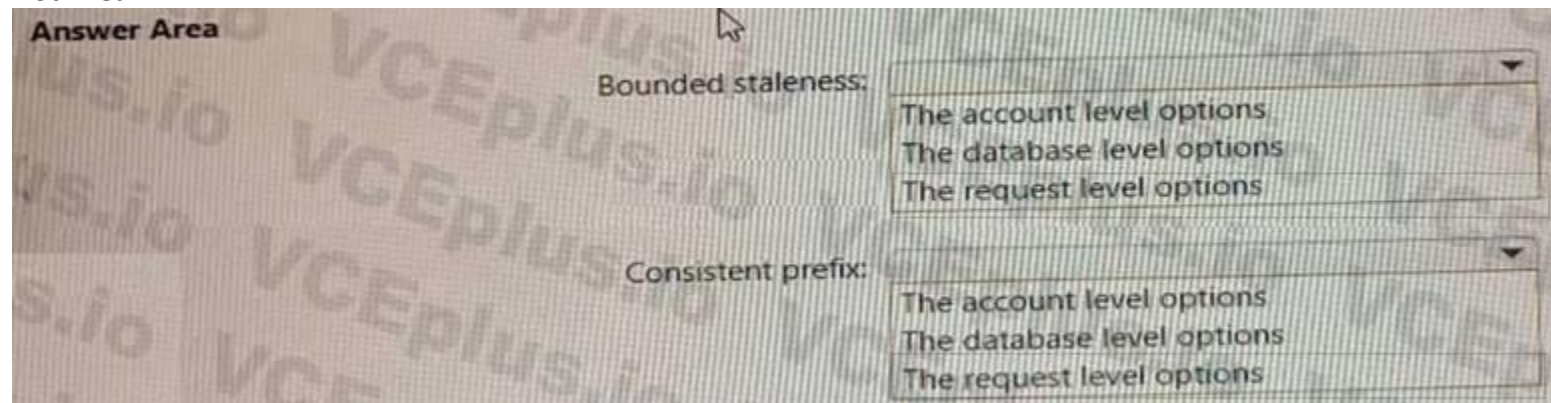
You have an app named App1.

You need to ensure that the read operations of App1 can request either bounded staleness or consistent prefix consistency.

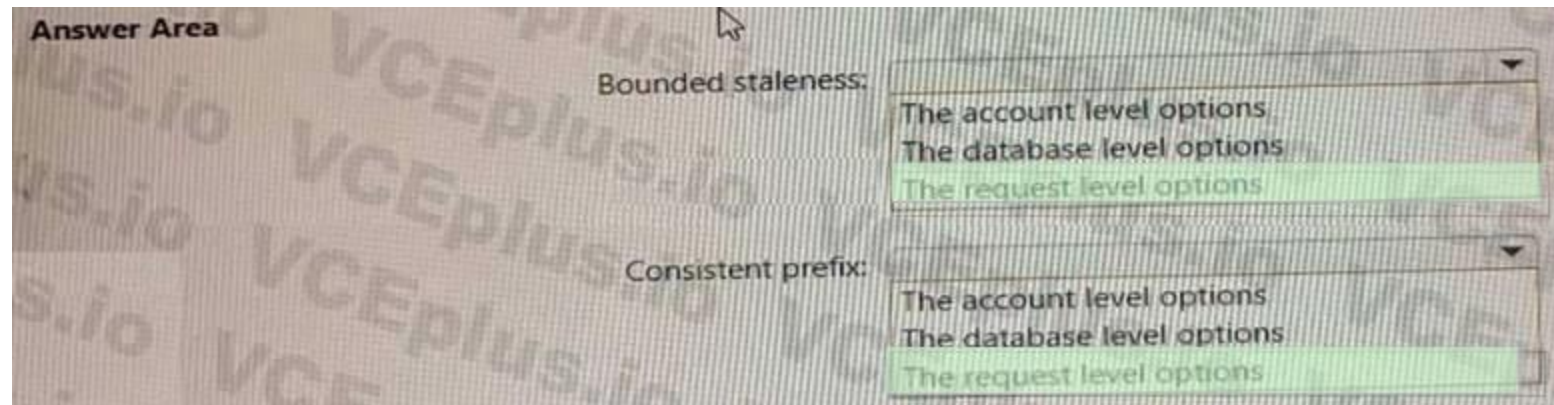
What should you modify for each consistency level? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:



Answer Area:



Section:

Explanation:

Box 1 = The request level options

Azure Cosmos DB offers five well-defined consistency levels: strong, bounded staleness, session, consistent prefix and eventual. You can configure the default consistency level on your Azure Cosmos DB account at any time².

The default consistency level applies to all databases and containers under that account¹. You can also override the default consistency level for a specific request by using the request options².

Box 2 = The request level options

To modify the consistency level of a read operation in Azure Cosmos DB, you can use request-level options to override the account's default consistency setting. Therefore, to ensure that the read operations of App1 can request either consistent prefix or session consistency, you need to modify the request-level options for each operation. Reference: - <https://docs.microsoft.com/enus/azure/cosmos-db/consistency-levels>

QUESTION 36

You have operational data in an Azure Cosmos DB for NoSQL database.

Database users report that the performance of the database degrades significantly when a business analytics team runs large Apache Spark-based queries against the database.

You need to reduce the impact that running the Spark-based queries has on the database users.

What should you implement?

- A. Azure Synapse Link
- B. a default consistency level of Consistent Prefix
- C. a default consistency level of Strong
- D. the Spark connector

Correct Answer: A

Section:

QUESTION 37

You plan to create an Azure Cosmos DB account that will use the NoSQL API.

You need to create a grouping strategy for items that will be stored in the account. The solution must ensure that write and read operations on the items can be performed within the same transact.

What should you use to group the items?

- A. logical partitions
- B. physical partitions
- C. databases
- D. containers

Correct Answer: A

Section:

QUESTION 38

HOTSPOT

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

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

Each application should use the weakest consistency level possible.

Which consistency level should you configure for each application? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

Hot Area:

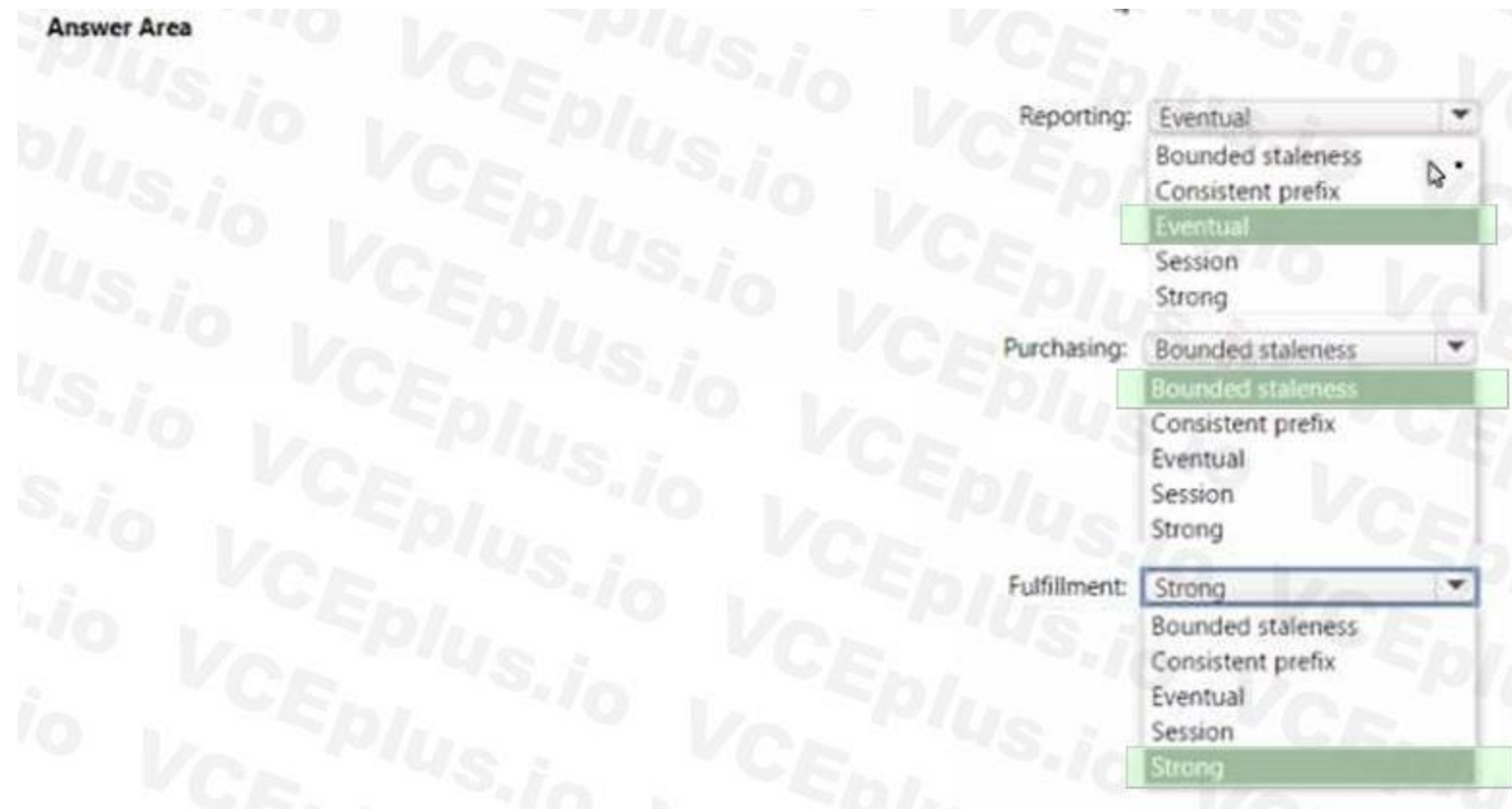
Answer Area

Reporting: Eventual
Bounded staleness
Consistent prefix
Eventual
Session
Strong

Purchasing: Bounded staleness
Bounded staleness
Consistent prefix
Eventual
Session
Strong

Fulfillment: Strong
Bounded staleness
Consistent prefix
Eventual
Session
Strong

Answer Area:



Section:

Explanation:

QUESTION 39

You have a global ecommerce application that stores data in an Azure Cosmos DB for NoSQL account.

The account is configured for multi-region writes.

You need to create a stored procedure for a custom conflict resolution policy for a new container. In the event of a conflict caused by a deletion the deletion must always take priority.

Which parameter should you check in the stored procedure function?

- A. conflictingItems
- B. is Tombstone
- C. existingitem
- D. incomingItem

Correct Answer: B

Section:

QUESTION 40

You plan to create an operational system that will store data in an Azure Cosmos DB for NoSQL account. You need to configure the account to meet the following requirements:

- Support Spar* queries.
- Support the analysis of data from the last six months.
- Only pay for analytical compute when running queries.

Which three actions should you perform? Each correct answer presents part of the solution. NOTE Each correct selection is worth one point.

- A. Create an Azure Synapse linked service.
- B. Create a container and set the time to live to six months.



- C. Create a container and set the analytical property to six months.
- D. Create an Azure Synapse pipeline.
- E. Create an Azure Databricks notebook.
- F. Enable Azure Synapse Link for the account

Correct Answer: C, E, F

Section:

QUESTION 41

HOTSPOT

You have an Azure Cosmos DB for NoSQL database named db1.

You run the following code:

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

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

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

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

NOTE: Each correct selection is worth one point.



Hot Area:

Answer Area

| Statements | Yes | No |
|--|-----------------------|-----------------------|
| Items inserted into the <code>customers</code> container without a <code>ttl</code> value will never expire. | <input type="radio"/> | <input type="radio"/> |
| All Items inserted into the <code>customers</code> container must include a <code>customerId</code> field. | <input type="radio"/> | <input type="radio"/> |
| Items inserted into the <code>customers</code> container that have a TTL value of 12 will expire in 12 days. | <input type="radio"/> | <input type="radio"/> |

Answer Area:

Answer Area

| Statements | Yes | No |
|--|----------------------------------|----------------------------------|
| Items inserted into the <code>customers</code> container without a <code>ttl</code> value will never expire. | <input checked="" type="radio"/> | <input type="radio"/> |
| All Items inserted into the <code>customers</code> container must include a <code>customerId</code> field. | <input checked="" type="radio"/> | <input type="radio"/> |
| Items inserted into the <code>customers</code> container that have a TTL value of 12 will expire in 12 days. | <input type="radio"/> | <input checked="" type="radio"/> |

Section:

Explanation:

QUESTION 42

HOTSPOT

You have a container named contained in an Azure Cosmos DB for NoSQL account. The following is a sample of a document in container1.

```
{
  "studentId": "631282"
  "firstName": "James",
  "lastName": "Smith",
  "enrollmentYear": 1990,
  "isActivelyEnrolled": true,
  "address": {
    "street": "",
    "city": "",
    "stateProvince": "",
    "postal": ""
  }
}
```

The container1 container has the following indexing policy.

```
{
  "indexingMode": "consistent",
  "includedPaths": [
    {
      "path": "/*"
    },
    {
      "path": "/address/city/?"
    }
  ],
  "excludedPaths": [
    {
      "path": "/address/*"
    },
    {
      "path": "/firstname/?"
    }
  ]
}
```

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



NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

| Statements | Yes | No |
|---|-----------------------|-----------------------|
| The <code>/isActivelyEnrolled</code> property is included in the index. | <input type="radio"/> | <input type="radio"/> |
| The <code>/firstname</code> property is included in the index. | <input type="radio"/> | <input type="radio"/> |
| The <code>/address/city</code> property is included in the index. | <input type="radio"/> | <input type="radio"/> |

Answer Area:

Answer Area

| Statements | Yes | No |
|---|----------------------------------|----------------------------------|
| The <code>/isActivelyEnrolled</code> property is included in the index. | <input checked="" type="radio"/> | <input type="radio"/> |
| The <code>/firstname</code> property is included in the index. | <input type="radio"/> | <input checked="" type="radio"/> |
| The <code>/address/city</code> property is included in the index. | <input checked="" type="radio"/> | <input type="radio"/> |

Section:

Explanation:

QUESTION 43

HOTSPOT

You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account named account1 and a Log Analytics workspace named Workspace1. Workspace 1 stores the logs of account1.

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

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

NOTE: Each correct selection is worth one point

Hot Area:

Answer Area

AzureDiagnostics
AzureActivity
AzureDiagnostics
AzureMetrics

```
| where ResourceProvider=="MICROSOFT.DOCUMENTDB" and Category==  
| where TimeGenerated >= ago(1d)
```

"DataPlaneRequests"
"ControlPlaneRequests"
"DataPlaneRequests"
"PartitionKeyStatistics"

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

Answer Area:

Answer Area

AzureDiagnostics
AzureActivity
AzureDiagnostics
AzureMetrics

```
| where ResourceProvider=="MICROSOFT.DOCUMENTDB" and Category==  
| where TimeGenerated >= ago(1d)
```

"DataPlaneRequests"
"ControlPlaneRequests"
"DataPlaneRequests"
"PartitionKeyStatistics"

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

Section:

Explanation:

QUESTION 44

TSPOT

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

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

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

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

What should you do? To answer, select the appropriate options in the answer are a. NOTE: Each correct selection is worth one point.

Hot Area:

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.

Answer Area:

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.

Section:

Explanation:

QUESTION 45

You have an Azure subscription.

You plan to create an Azure Cosmos DB for NoSQL database named DB1 that will store author and book data for authors that have each published up to ten books. Typical and frequent queries of the data will include:

- * All books written by an individual author
- * The synopsis of individual books

You need to recommend a data model for DB1. The solution must meet the following requirements:

- * Support transactional updates of the author and book data.
- * Minimize read operation costs.

What should you recommend?

- A. Create a single container that stores author items and book items, and then items that represent the relationship between the authors and their books.
- B. Create three containers, one that stores author items, a second that stores book items, and a third that stores items that represent the relationship between the authors and their books.
- C. Create two containers, one that stores author items and another that stores book items. Embed a list of each author's books in the corresponding author item.
- D. Create a single container that stores author items and book items. Embed a list of each author's books in the

Correct Answer: D

Section:

QUESTION 46

You have an Azure Cosmos DB for NoSQL account that contains a database named DB1 and a container named Container1. You need to manage the account by using the Azure Cosmos DB SDK. What should you do?

- A. List the physical partitions of Container1.
- B. Read a stored procedure in Container1.
- C. Create a user defined function (UDF) in Container1.
- D. Create a container in DB1.

Correct Answer: C

Section:

