



**Website:** <https://vceplus.com> - <https://vceplus.co> - <https://vceplus.io>



## Topic 1, Case Study 1

### Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study.

Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

### Overview

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

Existing environment. Cloud and Data Service Environments.

Litware has an Azure subscription that contains the resources shown in the following table.



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

Each container in productdb is configured for manual throughput.

The con-product container stores the company's product catalog data. Each document in con-product includes a conproductvendor value. Most queries targeting the data in con-product are in the following format.

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

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

Litware identifies the following issues:

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

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

Requirements. Planned Changes

Litware plans to implement a new Azure Cosmos DB Core (SQL) API account named account2 that will contain a database named iotdb. The iotdb database will contain two containers named con-iot1 and con-iot2.

Litware plans to make the following changes:

Store the telemetry data in account2.

Configure account1 to support multiple read-write regions.

Implement referential integrity for the con-product container.

Use Azure Functions to send notifications about product updates to different recipients.

Develop an app named App1 that will run from all locations and query the data in account1.

Develop an app named App2 that will run from the retail stores and query the data in account2. App2 must be limited to a single DNS endpoint when accessing account2. Requirements. Business Requirements Litware identifies the following business requirements:

Whenever there are multiple solutions for a requirement, select the solution that provides the best performance, as long as there are no additional costs associated. Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.

Minimize the number of firewall changes in the retail stores.

Requirements. Product Catalog Requirements

Litware identifies the following requirements for the product catalog:

Implement a custom conflict resolution policy for the product catalog data.

Minimize the frequency of errors during updates of the con-product container.

Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.

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

#### QUESTION 1

You are troubleshooting the current issues caused by the application updates.

Which action can address the application updates issue without affecting the functionality of the application?

- A. Enable time to live for the con-product container.
- B. Set the default consistency level of account1 to strong.
- C. Set the default consistency level of account1 to bounded staleness.
- D. Add a custom indexing policy to the con-product container.

ANSWER: C

Explanation:

Bounded staleness is frequently chosen by globally distributed applications that expect low write latencies but require total global order guarantee. Bounded staleness is great for applications featuring group collaboration and sharing, stock ticker, publish-subscribe/queueing etc.



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

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

#### QUESTION 2

You need to select the partition key for con-iot1. The solution must meet the IoT telemetry requirements.

What should you select?

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

ANSWER: D

Explanation:

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

Scenario: The iotdb database will contain two containers named con-iot1 and con-iot2. Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.

Reference:

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

#### Topic 2, Case Study 2

##### Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study.

Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

##### Overview

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

Existing environment. Cloud and Data Service Environments.

Litware has an Azure subscription that contains the resources shown in the following table.

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

Each container in productdb is configured for manual throughput.

The con-product container stores the company's product catalog data. Each document in con-product includes a conproductvendor value. Most queries targeting the data in con-product are in the following format.

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

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

Litware identifies the following issues:

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

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

Requirements. Planned Changes

Litware plans to implement a new Azure Cosmos DB Core (SQL) API account named account2 that will contain a database named iotdb. The iotdb database will contain two containers named con-iot1 and con-iot2.

Litware plans to make the following changes:

Store the telemetry data in account2.

Configure account1 to support multiple read-write regions.

Implement referential integrity for the con-product container.

Use Azure Functions to send notifications about product updates to different recipients.

Develop an app named App1 that will run from all locations and query the data in account1.

Develop an app named App2 that will run from the retail stores and query the data in account2. App2 must be limited to a single DNS endpoint when accessing account2. Requirements. Business Requirements Litware identifies the following business requirements:

Whenever there are multiple solutions for a requirement, select the solution that provides the best performance, as long as there are no additional costs associated. Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.

Minimize the number of firewall changes in the retail stores.

Requirements. Product Catalog Requirements

Litware identifies the following requirements for the product catalog:

Implement a custom conflict resolution policy for the product catalog data.

Minimize the frequency of errors during updates of the con-product container.

Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.

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

#### QUESTION 3

You configure multi-region writes for account1.

You need to ensure that App1 supports the new configuration for account1. The solution must meet the business requirements and the product catalog requirements.

What should you do?

- A. Set the default consistency level of account1 to bounded staleness.
- B. Create a private endpoint connection.
- C. Modify the connection policy of App1.
- D. Increase the number of request units per second (RU/s) allocated to the con-product and con-productVendor containers.

ANSWER: D

Explanation:

App1 queries the con-product and con-productVendor containers.

Note: Request unit is a performance currency abstracting the system resources such as CPU, IOPS, and memory that are required to perform the database operations supported by Azure Cosmos DB.

Scenario:

Develop an app named App1 that will run from all locations and query the data in account1.

Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.

Whenever there are multiple solutions for a requirement, select the solution that provides the best performance, as long as there are no additional costs associated.

Incorrect Answers:

A:

Bounded staleness relates to writes. App1 only do reads.

Note: Bounded staleness is frequently chosen by globally distributed applications that expect low write latencies but require total global order guarantee. Bounded staleness is great for applications featuring group collaboration and sharing, stock ticker, publish-subscribe/queueing etc.

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

#### QUESTION 4

You need to provide a solution for the Azure Functions notifications following updates to con-product. The solution must meet the business requirements and the product catalog requirements.

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

NOTE: Each correct selection is worth one point.

- A. Configure the trigger for each function to use a different leaseCollectionPrefix
- B. Configure the trigger for each function to use the same leaseCollectionName

C. Configure the trigger for each function to use a different leaseCollectionName

D. Configure the trigger for each function to use the same leaseCollectionPrefix

ANSWER: A B

Explanation: leaseCollectionPrefix: when set, the value is added as a prefix to the leases created in the Lease collection for this Function.

Using a prefix allows two separate Azure Functions to share the same Lease collection by using different prefixes.

Scenario: Use Azure Functions to send notifications about product updates to different recipients.

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

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

Topic 3, Mixed Questions

QUESTION 5

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

You discover that every query consumes request units (RUs) instead of using the cache.

You verify the IntegratedCacheItemHitRate metric and the IntegratedCacheQueryHitRate metric. Both metrics have values of 0.

You verify that the dedicated gateway cluster is provisioned and used in the connection string.

You need to ensure that App1 uses the Azure Cosmos DB integrated cache.

What should you configure?

A. the indexing policy of the Azure Cosmos DB container

B. the consistency level of the requests from App1

C. the connectivity mode of the App1 CosmosClient

D. the default consistency level of the Azure Cosmos DB account

ANSWER: C

Explanation:

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

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

QUESTION 6

HOTSPOT

You provision Azure resources by using the following Azure Resource Manager (ARM) template.



```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "db": {
      "defaultValue": "[resourceId('Microsoft.DocumentDB/databaseAccounts', 'prod1')]",
      "type": "String"
    },
    "sms": {
      "defaultValue": "[resourceId('Microsoft.Insights/actionGroups', 'sms')]",
      "type": "String"
    }
  },
  "variables": {},
  "resources": [
    {
      "type": "microsoft.insights/actionGroups",
      "apiVersion": "2019-06-01",
      "name": "sms",
      "location": "Global",
      "properties": {
        "groupShortName": "Send message",
        "enabled": true,
        "emailReceivers": [],
        "smsReceivers": [
          {
            "name": "Action-SMS",
            "countryCode": "44",
            "phoneNumber": "7111111111"
          }
        ]
      }
    },
    {
      "type": "microsoft.insights/activityLogAlerts",
      "apiVersion": "2020-10-01",
      "name": "Alert1",
      "location": "Global",
      "dependsOn": ["sms"],
      "properties": {
        "scopes": [ "[parameters('db')]" ],
        "condition": {
          "allOf": [
            {
              "field": "category",
              "equals": "Administrative"
            },
            {
              "field": "operationName",
              "equals": "Microsoft.DocumentDB/databaseAccounts/regenerateKey/action"
            }
          ]
        },
        "actions": {
          "actionGroups": [
            {
              "actionGroupId": "[parameters('sms')]",
              "webhookProperties": {}
            }
          ]
        },
        "enabled": true
      }
    }
  ]
}
```

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 alert will be triggered when an Azure Cosmos DB key is used	<input type="radio"/>	<input type="radio"/>
	Two alert actions will be performed when the alert is triggered	<input type="radio"/>	<input type="radio"/>
	The alert will be triggered when an item that has a new partition key value is created	<input type="radio"/>	<input type="radio"/>

ANSWER:

Statements	Yes	No
The alert will be triggered when an Azure Cosmos DB key is used	<input type="radio"/>	<input checked="" type="radio"/>
Two alert actions will be performed when the alert is triggered	<input type="radio"/>	<input checked="" type="radio"/>
The alert will be triggered when an item that has a new partition key value is created	<input checked="" type="radio"/>	<input type="radio"/>

Explanation:

Box 1: No

An alert is triggered when the DB key is regenerated, not when it is used.

Note: The az cosmosdb keys regenerate command regenerates an access key for a Azure Cosmos DB database account.

Box 2: No

Only an SMS action will be taken.

Emailreceivers is empty so no email action is taken.

Box 3: Yes

Yes, an alert is triggered when the DB key is regenerated.

Reference: <https://docs.microsoft.com/en-us/cli/azure/cosmosdb/keys>

QUESTION 7

HOTSPOT

You have an Azure Cosmos DB Core (SQL) API account named account1 that has the disableKeyBasedMetadataWriteAccess property enabled.

You are developing an app named App1 that will be used by a user named DevUser1 to create containers in account1.

DevUser1 has a non-privileged user account in the Azure Active Directory (Azure AD) tenant.

You need to ensure that DevUser1 can use App1 to create containers in account1.

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

NOTE: Each correct selection is worth one point.

Hot Area:



**Answer Area**

Grant permissions to create containers by using:

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

Create containers by using the:

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

ANSWER:

**Answer Area**

Grant permissions to create containers by using:

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

Create containers by using the:

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

Explanation:

Box 1: Resource tokens

Resource tokens provide access to the application resources within a database. Resource tokens:

Provide access to specific containers, partition keys, documents, attachments, stored procedures, triggers, and UDFs.

Box 2: Azure Resource Manager API

You can use Azure Resource Manager to help deploy and manage your Azure Cosmos DB accounts, databases, and containers.

Incorrect Answers:

The Microsoft Graph API is a RESTful web API that enables you to access Microsoft Cloud service resources.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/secure-access-to-data> <https://docs.microsoft.com/enus/rest/api/resources/>

QUESTION 8

HOTSPOT

You have an Azure Cosmos DB Core (SQL) account that has a single write region in West Europe.

You run the following Azure CLI script.

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

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

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

NOTE: Each correct selection is worth one point.

Hot Area:

Statements	Yes	No
After running the script, there will be an instance of Azure Cosmos DB in North Europe that is writable	<input type="radio"/>	<input type="radio"/>
After running the script, the Azure Cosmos DB instance in West Europe will be writable	<input type="radio"/>	<input type="radio"/>
The cost of the Azure Cosmos DB account is unaffected by running the script	<input type="radio"/>	<input type="radio"/>

ANSWER:

Statements	Yes	No
After running the script, there will be an instance of Azure Cosmos DB in North Europe that is writable	<input checked="" type="radio"/>	<input type="radio"/>
After running the script, the Azure Cosmos DB instance in West Europe will be writable	<input type="radio"/>	<input checked="" type="radio"/>
The cost of the Azure Cosmos DB account is unaffected by running the script	<input checked="" type="radio"/>	<input type="radio"/>

Explanation:

Box 1: Yes

The Automatic failover option allows Azure Cosmos DB to failover to the region with the highest failover priority with no user action should a region become unavailable.

Box 2: No

West Europe is used for failover. Only North Europe is writable.

To Configure multi-region set UseMultipleWriteLocations to true.

Box 3: Yes

Provisioned throughput with single write region costs \$0.008/hour per 100 RU/s and provisioned throughput with multiple writable regions costs \$0.016/per hour per 100 RU/s.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-multi-master> <https://docs.microsoft.com/en-us/azure/cosmosdb/optimize-cost-regions>

## QUESTION 9

You are developing an application that will use an Azure Cosmos DB Core (SQL) API account as a data source.

You need to create a report that displays the top five most ordered fruits as shown in the following table.

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

A collection that contains aggregated data already exists. The following is a sample document:

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

Which two queries can you use to retrieve data for the report? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. 

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

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

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

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

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: B D

Explanation:

ARRAY\_CONTAINS returns a Boolean indicating whether the array contains the specified value. You can check for a partial or full match of an object by using a boolean expression within the command.

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>

QUESTION 10

HOTSPOT

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

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

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

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

NOTE: Each correct selection is worth one point.

Hot Area:

### 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

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

QUESTION 11

HOTSPOT

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account. The container1 container has 120 GB of data.

The following is a sample of a document in container1.

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

The orderId property is used as the partition key.

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

NOTE: Each correct selection is worth one point.

Hot Area:

Statements	Yes	No
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.orderDate = "2019-05-03"</pre>	<input type="radio"/>	<input type="radio"/>
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.customerId = "5425"</pre>	<input type="radio"/>	<input type="radio"/>
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.orderDate = "2019-05-03" and c.orderId = "9d7816e6-f401-42ba-ad05-0e03de35c0b8"</pre>	<input type="radio"/>	<input type="radio"/>

ANSWER:

Statements	Yes	No
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.orderDate = "2019-05-03"</pre>	<input checked="" type="radio"/>	<input type="radio"/>
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.customerId = "5425"</pre>	<input checked="" type="radio"/>	<input type="radio"/>
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.orderDate = "2019-05-03" and c.orderId = "9d7816e6-f401-42ba-ad05-0e03de35c0b8"</pre>	<input type="radio"/>	<input checked="" type="radio"/>

Explanation:

Box 1: Yes

Records with different OrderIDs will match.

Box 2: Yes

Records with different OrderIDs will match.

Box 3: No

Only records with one specific OrderId will match

#### QUESTION 12

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

The following is a sample of the data.

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

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

Minimizes the partition skew

Avoids capacity limits Avoids hot partitions

What should you do?

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

ANSWER: D

Explanation:

Use a partition key with a random suffix. Distribute the workload more evenly is to append a random number at the end of the partition key value. When you distribute items in this way, you can perform parallel write operations across partitions.

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>

#### QUESTION 13

You maintain a relational database for a book publisher. The database contains the following tables.

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



The most common query lists the books for a given authorId.

You need to develop a non-relational data model for Azure Cosmos DB Core (SQL) API that will replace the relational database. The solution must minimize latency and read operation costs.

What should you include in the solution?

- A. Create a container for Author and a container for Book. In each Author document, embed bookId for each book by the author. In each Book document embed authorId of each author.
- B. Create Author, Book, and BookauthorInk documents in the same container.
- C. Create a container that contains a document for each Author and a document for each Book. In each Book document, embed authorId.
- D. Create a container for Author and a container for Book. In each Author document and Book document embed the data from BookauthorInk.

ANSWER: A

Explanation:

Store multiple entity types in the same container.

QUESTION 14

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

You run the following query against a container in the account.

SELECT

IS\_NUMBER("1234") AS A,  
IS\_NUMBER(1234) AS B,  
IS\_NUMBER({prop: 1234}) AS C

What is the output of the query?

- A. [{"A": false, "B": true, "C": false}]
- B. [{"A": true, "B": false, "C": true}]
- C. [{"A": true, "B": true, "C": false}]
- D. [{"A": true, "B": true, "C": true}]

ANSWER: A

Explanation:

IS\_NUMBER returns a Boolean value indicating if the type of the specified expression is a number. "1234" is a string, not a number.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/sql-query-is-number>

#### QUESTION 15

You need to implement a trigger in Azure Cosmos DB Core (SQL) API that will run before an item is inserted into a container.

Which two actions should you perform to ensure that the trigger runs? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

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

ANSWER: C

Explanation:

C: When triggers are registered, you can specify the operations that it can run with.

F: When executing, pre-triggers are passed in the RequestOptions object by specifying PreTriggerInclude and then passing the name of the trigger in a List object.

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

#### QUESTION 16

##### HOTSPOT

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

You need to use the Azure Cosmos DB SDK to replace a document by using optimistic concurrency.

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

NOTE: Each correct selection is worth one point.

Hot Area:

**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>

#### QUESTION 17

##### HOTSPOT

You are creating a database in an Azure Cosmos DB Core (SQL) API account. The database will be used by an application that will provide users with the ability to share online posts. Users will also be able to submit comments on other users' posts.

You need to store the data shown in the following table.

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

The application has the following characteristics:

Users can submit an unlimited number of posts.

The average number of posts submitted by a user will be more than 1,000.

Posts can have an unlimited number of comments from different users.

The average number of comments per post will be 100, but many posts will exceed 1,000 comments. Users will be limited to having a maximum of 20 interests.

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

NOTE: Each correct selection is worth one point.

Hot Area:

**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	<input type="radio"/>	<input type="radio"/>
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	<input type="radio"/>	<input type="radio"/>
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	<input type="radio"/>	<input type="radio"/>

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	<input checked="" type="radio"/>	<input type="radio"/>
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	<input checked="" type="radio"/>	<input type="radio"/>
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	<input type="radio"/>	<input checked="" type="radio"/>

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.

## QUESTION 18


### DRAG DROP

You have an app that stores data in an Azure Cosmos DB Core (SQL) API account. The app performs queries that return large result sets.

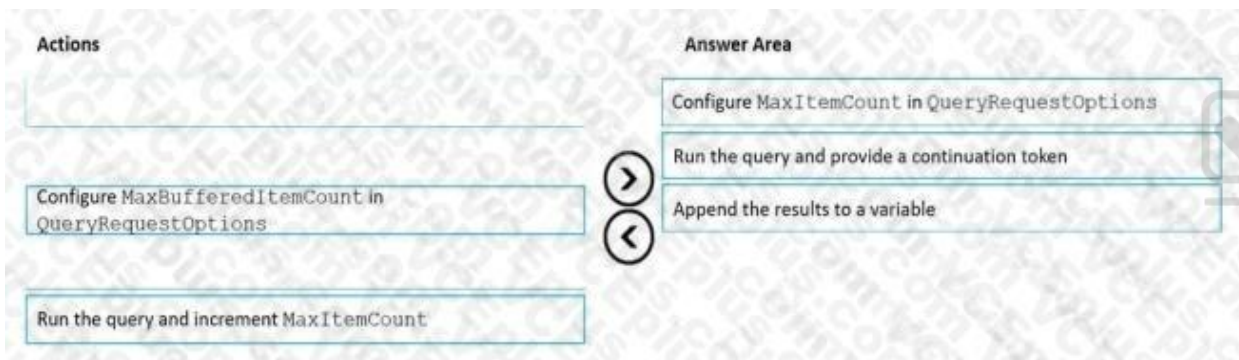
You need to return a complete result set to the app by using pagination. Each page of results must return 80 items.

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.

Select and Place:



ANSWER:



Explanation:

Step 1: Configure the MaxItemCount in QueryRequestOptions

You can specify the maximum number of items returned by a query by setting the MaxItemCount. The MaxItemCount is specified per request and tells the query engine to return that number of items or fewer.

Box 2: Run the query and provide a continuation token

In the .NET SDK and Java SDK you can optionally use continuation tokens as a bookmark for your query's progress. Azure Cosmos DB query executions are stateless at the server side and can be resumed at any time using the continuation token.

If the query returns a continuation token, then there are additional query results. Step 3: Append the results to a variable

Reference:

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

## QUESTION 19

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

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

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

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

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

Does this meet the goal?

A. Yes

B. No

ANSWER: A

Explanation:

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

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

Reference:

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

#### QUESTION 20

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

ANSWER: B

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 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 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

ANSWER: B

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 22

You need to identify which connectivity mode to use when implementing App2. The solution must support the planned changes and meet the business requirements.

Which connectivity mode should you identify?

A. Direct mode over HTTPS

B. Gateway mode (using HTTPS)

C. Direct mode over TCP

ANSWER: C

Explanation:

Scenario: Develop an app named App2 that will run from the retail stores and query the data in account2. App2 must be limited to a single DNS endpoint when accessing account2.

By using Azure Private Link, you can connect to an Azure Cosmos account via a private endpoint. The private endpoint is a set of private IP addresses in a subnet within your virtual network.

When you're using Private Link with an Azure Cosmos account through a direct mode connection, you can use only the TCP protocol. The HTTP protocol is not currently supported.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-configure-private-endpoints>

#### QUESTION 23

##### DRAG DROP

You have an Azure Cosmos DB Core (SQL) API account that is configured for multi-region writes. The account contains a database that has two containers named container1 and container2.

The following is a sample of a document in container1:

```
{
```

```
"customerId": 1234,
"firstName": "John",
"lastName": "Smith",
"policyYear": 2021 }
```

The following is a sample of a document in container2:

```
{
"gpsId": 1234,
"latitude": 38.8951,
"longitude": -77.0364 }
```

You need to configure conflict resolution to meet the following requirements:

For container1 you must resolve conflicts by using the highest value for policyYear.

For container2 you must resolve conflicts by accepting the distance closest to latitude: 40.730610 and longitude: - 73.935242. Administrative effort must be minimized to implement the solution.

What should you configure for each container? To answer, drag the appropriate configurations to the correct containers.

Each configuration may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Select and Place:

**Configurations**

Last Write Wins (default) mode

Merge Procedures (custom) mode

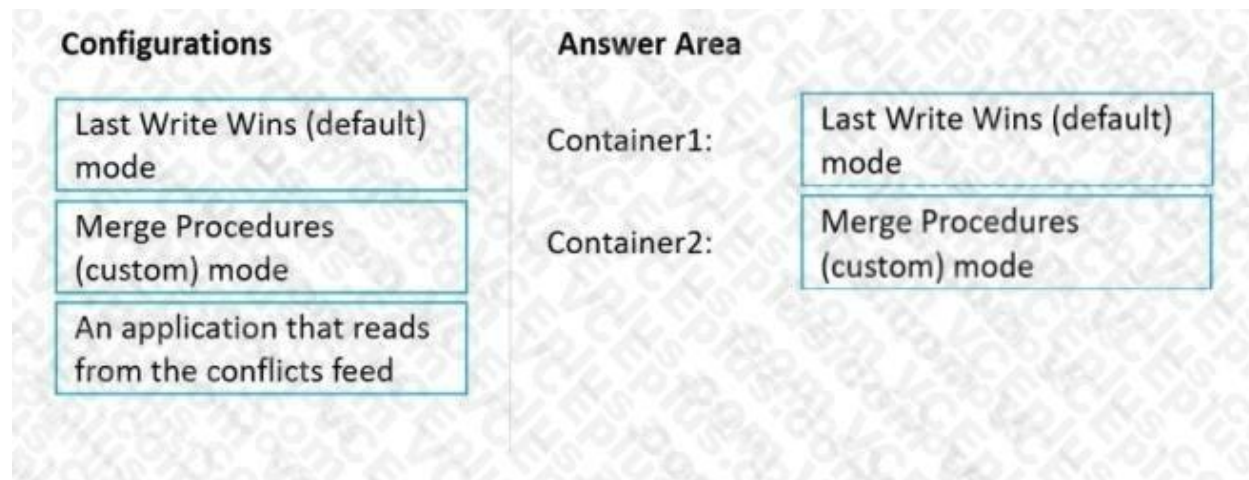
An application that reads from the conflicts feed

**Answer Area**

Container1:

Container2:

ANSWER:



Explanation:

Box 1: Last Write Wins (LWW) (default) mode

Last Write Wins (LWW): This resolution policy, by default, uses a system-defined timestamp property. It's based on the timesynchronization clock protocol.

Box 2: Merge Procedures (custom) mode

Custom: This resolution policy is designed for application-defined semantics for reconciliation of conflicts. When you set this policy on your Azure Cosmos container, you also need to register a merge stored procedure. This procedure is automatically invoked when conflicts are detected under a database transaction on the server. The system provides exactly once guarantee for the execution of a merge procedure as part of the commitment protocol.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/conflict-resolution-policies> <https://docs.microsoft.com/enus/azure/cosmos-db/sql/how-to-manage-conflicts>

#### QUESTION 24

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

Approximately 5,000 IoT devices submit data every five minutes by using the same telemetryId value.

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

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

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

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

What should you do?

- A. Decrease the offerThroughput value for the container.
- B. Increase the offerThroughput value for the container.

C. Move the data to a new container that has a partition key of deviceId.

D. Move the data to a new container that uses a partition key of date.

ANSWER: C

Explanation:

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

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

## QUESTION 25

### HOTSPOT

You have an Azure Cosmos DB Core (SQL) API account named storage1 that uses provisioned throughput capacity mode.

The storage1 account contains the databases shown in the following table.

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

The databases contain the containers shown in the following table.

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

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

NOTE: Each correct selection is worth one point.

Hot Area:

Statements	Yes	No
At a minimum, you will be billed for 4,000 RU/s per hour for db1	<input type="radio"/>	<input type="radio"/>
The maximum throughput that can be consumed by cn11 is 400 RU/s	<input type="radio"/>	<input type="radio"/>
To db2, you can add a new container that uses database throughput	<input type="radio"/>	<input type="radio"/>

ANSWER:

Statements	Yes	No
At a minimum, you will be billed for 4,000 RU/s per hour for db1	<input type="radio"/>	<input checked="" type="radio"/>
The maximum throughput that can be consumed by cn11 is 400 RU/s	<input type="radio"/>	<input checked="" type="radio"/>
To db2, you can add a new container that uses database throughput	<input checked="" type="radio"/>	<input type="radio"/>

Explanation:

Box 1: No

Four containers with 1000 RU/s each.

Box 2: No

Max 8000 RU/s for db2. 8 containers, so 1000 RU/s for each container.

Box 3: Yes

Max 8000 RU/s for db2. 8 containers, so 1000 RU/s for each container. Can very well add an additional container.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/plan-manage-costs> <https://azure.microsoft.com/enus/pricing/details/cosmos-db/>

QUESTION 26

HOTSPOT

You have a database named telemetry in an Azure Cosmos DB Core (SQL) API account that stores IoT data. The database contains two containers named readings and devices.

Documents in readings have the following structure. id deviceid timestamp ownerid measures (array)

- type
- value
- metricid

Documents in devices have the following structure.

id deviceid owner

- ownerid
- emailaddress
- name brand model

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

NOTE: Each correct selection is worth one point.

Hot Area:

Statements	Yes	No
To return for all devices owned by a specific emailaddress, multiple queries must be performed	<input type="radio"/>	<input type="radio"/>
To return deviceid, ownerid, timestamp, and value for a specific metricid, a join must be performed	<input type="radio"/>	<input type="radio"/>
To return deviceid, ownerid, emailaddress, and model, a join must be performed	<input type="radio"/>	<input type="radio"/>

ANSWER:

Statements	Yes	No
To return for all devices owned by a specific emailaddress, multiple queries must be performed	<input checked="" type="radio"/>	<input type="radio"/>
To return deviceid, ownerid, timestamp, and value for a specific metricid, a join must be performed	<input type="radio"/>	<input checked="" type="radio"/>
To return deviceid, ownerid, emailaddress, and model, a join must be performed	<input type="radio"/>	<input checked="" type="radio"/>

Explanation:

Box 1: Yes

Need to join readings and devices.

Box 2: No

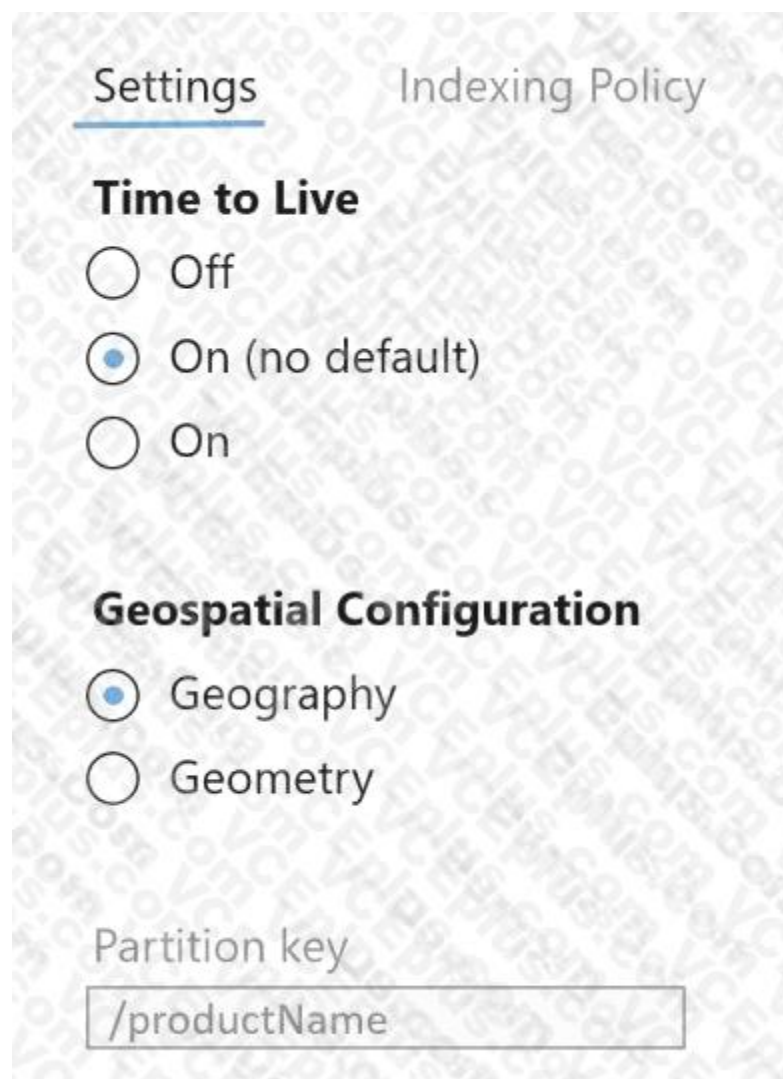
Only readings is required. All required fields are in readings.

Box 3: No

Only devices is required. All required fields are in devices.

#### QUESTION 27

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 an Azure Cosmos DB container. Under 'Time to Live', the 'On (no default)' option is selected. Under 'Geospatial Configuration', the 'Geography' option is selected. 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.

ANSWER: C

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 28

You have an Azure Cosmos DB Core (SQL) API account that uses a custom conflict resolution policy. The account has a registered merge procedure that throws a runtime exception.

The runtime exception prevents conflicts from being resolved.

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

What should you use?

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

ANSWER: D

Explanation:

The Azure Cosmos DB Trigger uses the Azure Cosmos DB Change Feed to listen for inserts and updates across partitions.

The change feed publishes inserts and updates, not deletions.

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

#### QUESTION 29

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.

ANSWER: B

Explanation:

You can facilitate aggregate data by using Change Feed and Azure Functions, and then use it for reporting.

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

QUESTION 30

HOTSPOT

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

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

You have the following Azure functions:

A function named Fn1 that reads the change feed of cn1

A function named Fn2 that reads the change feed of cn2

A function named Fn3 that reads the change feed of cn3

You perform the following actions:

Delete an item named item1 from cn1.

Update an item named item2 in cn2.

For an item named item3 in cn3, update the item time to live to 3,600 seconds.

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

NOTE: Each correct selection is worth one point.

Hot Area:

Statements	Yes	No
Fn1 will receive item1 from the change feed	<input type="radio"/>	<input type="radio"/>
Fn2 can check the _etag of item2 to see whether the item is an update or an insert	<input type="radio"/>	<input type="radio"/>
Fn3 will receive item3 from the change feed	<input type="radio"/>	<input type="radio"/>

ANSWER:

Statements	Yes	No
Fn1 will receive item1 from the change feed	<input type="radio"/>	<input checked="" type="radio"/>
Fn2 can check the _etag of item2 to see whether the item is an update or an insert	<input type="radio"/>	<input checked="" type="radio"/>
Fn3 will receive item3 from the change feed	<input checked="" type="radio"/>	<input type="radio"/>

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>

QUESTION 31

HOTSPOT

You configure Azure Cognitive Search to index a container in an Azure Cosmos DB Core (SQL) API account as shown in the following exhibit.



Field Name	Type	Retrievable	Filterable	Sortable	Facetable	Searchable	Analyzer	Suggester
id	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
name	Edm.String	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Standard - Lucene	...
headquarters	Edm.ComplexType							...
country	Edm.String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
iso	Edm.String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
employees	Edm.Int32	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			...
...	...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...

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**

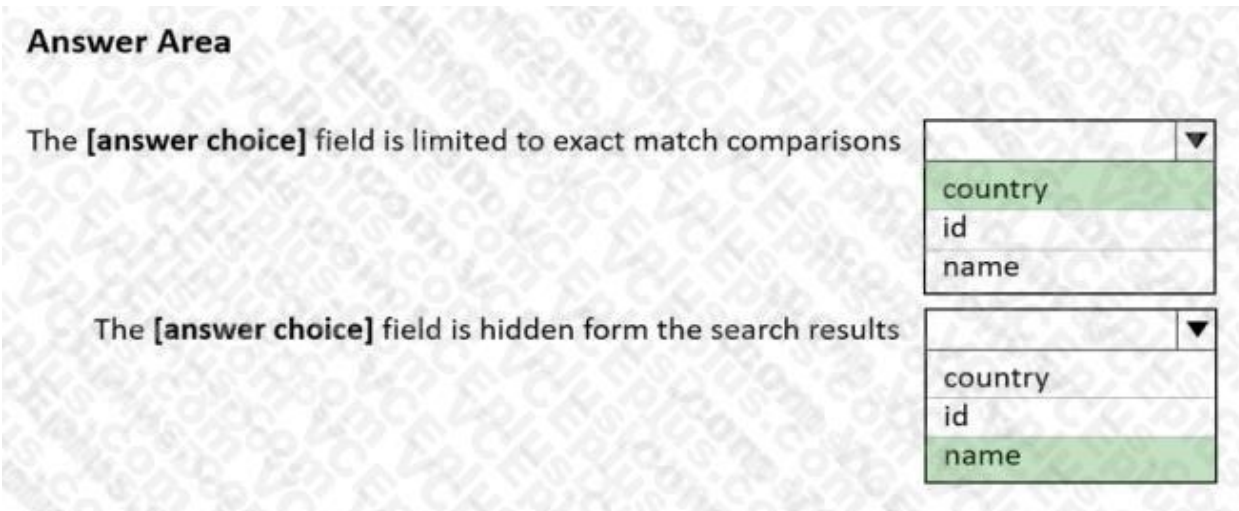
The [answer choice] field is limited to exact match comparisons

The [answer choice] field is hidden form the search results

country  
id  
name

country  
id  
name

ANSWER:



Explanation:

Box 1: country

The country field is filterable.

Note: filterable: Indicates whether to enable the field to be referenced in \$filter queries. Filterable differs from searchable in how strings are handled. Fields of type Edm.String or Collection(Edm.String) that are filterable do not undergo lexical analysis, so comparisons are for exact matches only.

Box 2: name

The name field is not Retrievable.

Retrieval: Indicates whether the field can be returned in a search result. Set this attribute to false if you want to use a field (for example, margin) as a filter, sorting, or scoring mechanism but do not want the field to be visible to the end user.

Note: searchable: Indicates whether the field is full-text searchable and can be referenced in search queries.

Reference: <https://docs.microsoft.com/en-us/rest/api/searchservice/create-index>

## QUESTION 32

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

ANSWER: B

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 33

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

ANSWER: B

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 34

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

ANSWER: A

Explanation:

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:



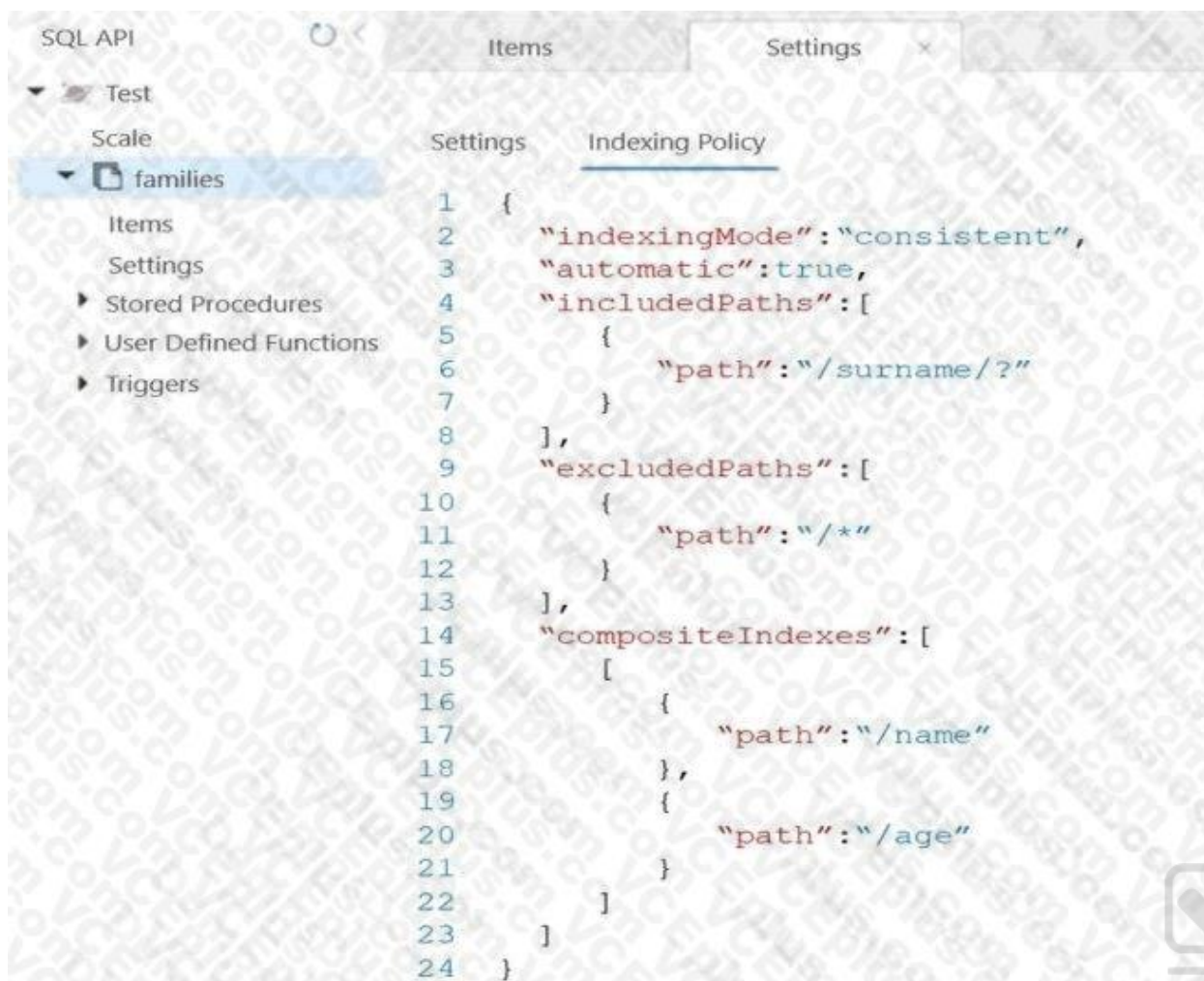
Reference:

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

QUESTION 35

HOTSPOT

You have the indexing policy shown in the following exhibit.



```

1  {
2    "indexingMode": "consistent",
3    "automatic": true,
4    "includedPaths": [
5      {
6        "path": "/surname/?"
7      }
8    ],
9    "excludedPaths": [
10     {
11       "path": "/*"
12     }
13   ],
14   "compositeIndexes": [
15     [
16       {
17         "path": "/name"
18       },
19       {
20         "path": "/age"
21       }
22     ]
23   ]
24 }

```

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

NOTE: Each correct selection is worth one point.

Hot Area:

**Answer Area**

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

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

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

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

ANSWER:

### Answer Area

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

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

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

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

Explanation:

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

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

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

If the composite index paths do not match the sequence of the properties in the ORDER BY clause, then the composite index can't support the query. The order of composite index paths (ascending or descending) should also match the order in the ORDER BY clause. The composite index also supports an ORDER BY clause with the opposite order on all paths.

Box 2: At the same time as the item creation

Azure Cosmos DB supports two indexing modes:

Consistent: The index is updated synchronously as you create, update or delete items. This means that the consistency of your read queries will be the consistency configured for the account. None: Indexing is disabled on the container.

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

### QUESTION 36

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

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

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

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

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

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

ANSWER: D

Explanation:

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

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

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>

#### QUESTION 37

##### HOTSPOT

You have a container named container1 in an Azure Cosmos DB Core (SQL) API 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",
  "includePaths": [
    {
      "path": "/*"
```



```

},
{
  "path": "/address/city/?"
}
],
"excludePaths": [
{
  "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:



Statements	Yes	No
The <code>/isActivelyEnrolled</code> property is included in the index	<input type="radio"/>	<input type="radio"/>
The <code>/fisrtname</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:

Answer Area	
Statements	Yes No
The /isActivelyEnrolled property is included in the index	<input checked="" type="radio"/> <input type="radio"/>
The /fisrtname property is included in the index	<input type="radio"/> <input checked="" type="radio"/>
The /address/city property is included in the index	<input checked="" type="radio"/> <input type="radio"/>

Explanation:

Box 1: Yes

"path": "/" is in includePaths.

Include the root path to selectively exclude paths that don't need to be indexed. This is the recommended approach as it lets Azure Cosmos DB proactively index any new property that may be added to your model.

Box 2: No

"path": "/firstName/?" is in excludePaths.

Box 3: Yes

"path": "/address/city/?" is in includePaths

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



#### QUESTION 38

You have the following query.

SELECT \* FROM ?

WHERE c.sensor = "TEMP1"

AND c.value >= 22

AND c.timestamp >= 1619146031231

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

What should you recommend?

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

ANSWER: A

Explanation:

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

Consider the following query:

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

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

Queries with multiple equality filters and a maximum of one range filter (such as `c.name = 'Tim' and c.age >= 18`) will utilize the composite index.

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

#### QUESTION 39

##### HOTSPOT

You have a database in an Azure Cosmos DB SQL API Core (SQL) account that is used for development.

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

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

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

NOTE: Each correct selection is worth one point.

Hot Area:

### Answer Area

Backup interval

	▼
1 hour	
24 hours	
1 weeks	

Backup retention

	▼
2 days	
1 week	
30 days	

ANSWER:

**Answer Area**

Backup interval

▼

1 hour  
24 hours  
1 weeks

Backup retention

▼

2 days  
1 week  
30 days

Explanation:

QUESTION 40

HOTSPOT

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

You have the Azure virtual networks and subnets shown in the following table.

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

The vnet1 and vnet2 networks are connected by using a virtual network peer.

The Firewall and virtual network settings for account1 are configured as shown in the exhibit.

Allow access from  
☐ All networks ☒ Selected networks  
 Configure network security for your Azure Cosmos DB account. [Learn more.](#)

Virtual networks  
 Secure your Azure Cosmos DB account with virtual networks. [+ Add existing virtual network](#) [+Add new virtual network](#)

Virtual Network	Subnet	Address range	Endpoint Status
▼ vnet1	1	10.0.0.0/16	
	vnet1.subnet1	10.0.1.0/24	✓ Enabled

Firewall  
 Add IP ranges to allow access from the internet or your on-premises networks. [+Add my current IP](#) ⓘ

**IP(Single IPv4 or CIDR range)**

Exceptions  
☐ Accept connections from within public Azure datacenters ⓘ  
☒ Allow access from Azure Portal ⓘ

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
VM1 can access account 1	<input type="radio"/>	<input type="radio"/>
VM2 can access account 1	<input type="radio"/>	<input type="radio"/>
VM3 can access account 1	<input type="radio"/>	<input type="radio"/>

ANSWER:

Answer Area		
Statements	Yes	No
VM1 can access account 1	<input checked="" type="radio"/>	<input type="radio"/>
VM2 can access account 1	<input type="radio"/>	<input checked="" type="radio"/>
VM3 can access account 1	<input type="radio"/>	<input checked="" type="radio"/>

Explanation:

Box 1: Yes

VM1 is on vnet1.subnet1 which has the Endpoint Status enabled.

Box 2: No

Only virtual network and their subnets added to Azure Cosmos account have access. Their peered VNets cannot access the account until the subnets within peered virtual networks are added to the account.

Box 3: No

Only virtual network and their subnets added to Azure Cosmos account have access.

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

#### QUESTION 41

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

ANSWER: A B G

Explanation:

To Configure customer-managed keys for your Azure Cosmos account with Azure Key Vault: Add an access policy to your Azure Key Vault instance:

1. From the Azure portal, go to the Azure Key Vault instance that you plan to use to host your encryption keys. Select Access Policies from the left menu:



2. Select + Add Access Policy.

3. Under the Key permissions drop-down menu, select Get, Unwrap Key, and Wrap Key permissions:



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

#### QUESTION 42

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"

ANSWER: C D F

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/kafkaconnect-deep-dive-converters-serialization-explained/>

#### QUESTION 43

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

ANSWER: C

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 44

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 role-based 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

ANSWER: A

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 45

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"

ANSWER: A

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 46

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.

ANSWER: A

Explanation:

When creating a new Azure Cosmos DB account, in the Backup policy tab, choose continuous mode to enable the point in time restore functionality for the new account. With the point-in-time restore, data is restored to a new account, currently you can't restore to an existing account.

Home > Create a resource > Select API option >

## Create Azure Cosmos DB Account - Core (SQL)

Basics Global Distribution Networking **Backup Policy** Encryption Tags Review + create

Azure Cosmos DB provides two different backup policies. You will not be able to switch between backup policies after the account has been created. Learn more about the differences of the two backup policies and pricing details. [Learn more](#) >

Backup policy: ☐ Periodic ☒ Continuous

[Learn more about pricing and zone resiliency configurations related to continuous backup mode](#) >

[Review + create](#) [Previous](#) [Next: Encryption](#)

Provision an Azure Cosmos DB account with continuous backup configuration.

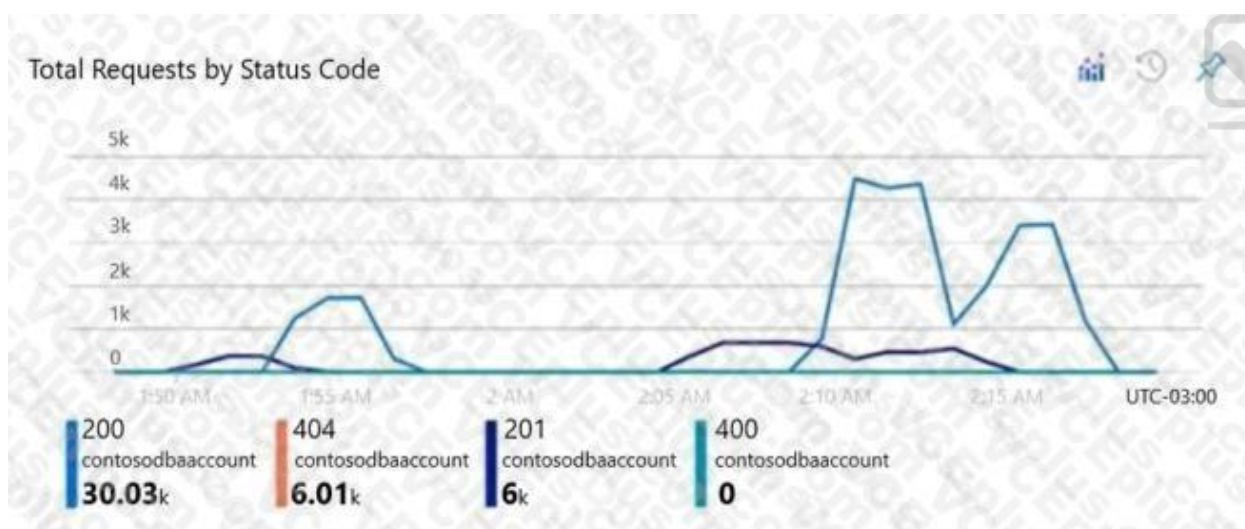
Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/provision-account-continuous-backup>

### QUESTION 47

#### HOTSPOT

You have an Azure Cosmos DB Core (SQL) API account used by an application named App1.

You open the Insights pane for the account and see the following chart.



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

NOTE: Each correct selection is worth one point.

Hot Area:

**Answer Area**

The HTTP 404 status code is caused by [answer choice]

incorrect connection URLs
an intermittent firewall issue
incorrectly formatted partition keys
requesting resources that do not exist

There are [answer choice] successful resource creations in the account during the time period of the chart

zero
6 thousand
6.01 thousand
30.03 thousand
36.03 thousand

ANSWER:

**Answer Area**

The HTTP 404 status code is caused by [answer choice]

incorrect connection URLs
an intermittent firewall issue
incorrectly formatted partition keys
requesting resources that do not exist

There are [answer choice] successful resource creations in the account during the time period of the chart

zero
6 thousand
6.01 thousand
30.03 thousand
36.03 thousand

Explanation:

Box 1: incorrect connection URLs

400 Bad Request: Returned when there is an error in the request URI, headers, or body. The response body will contain an error message explaining what the specific problem is.

The HyperText Transfer Protocol (HTTP) 400 Bad Request response status code indicates that the server cannot or will not process the request due to something that is perceived to be a client error (for example, malformed request syntax, invalid request message framing, or deceptive request routing).

Box 2: 6 thousand

201 Created: Success on PUT or POST. Object created or updated successfully.

Note:

200 OK: Success on GET, PUT, or POST. Returned for a successful response.

404 Not Found: Returned when a resource does not exist on the server. If you are managing or querying an index, check the syntax and verify the index name is specified correctly.

Reference:

<https://docs.microsoft.com/en-us/rest/api/searchservice/http-status-codes>

#### QUESTION 48

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 + ''"

ANSWER: C

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 49

##### HOTSPOT

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

Name	Description
development	<ul style="list-style-type: none"> <li>Supports the development of new application features</li> <li>Used intermittently as needed during development</li> </ul>
shipments	<ul style="list-style-type: none"> <li>Captures over 100,000 updates per second generated at unpredictable times throughout the business day</li> <li>Used with Azure Synapse Link for analytics</li> </ul>

How should you provision the containers within each account to minimize costs? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

**Answer Area**

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:

**Answer Area**

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 costprohibitive, 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 (&#x003C;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/cosmosdb/provision-throughput-autoscale#use-cases-of-autoscale>

### QUESTION 50

#### HOTSPOT

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

In account1, you run the following query in a container that contains 100GB of data.

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

You view the following metrics while performing the query.

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

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

NOTE: Each correct selection is worth one point.

Hot Area:

Statements	Yes	No
The query performs a cross-partition query	<input type="radio"/>	<input type="radio"/>
The query uses an index	<input type="radio"/>	<input type="radio"/>
Recreating the container with the partition key set to /categoryId will improve the performance of the query	<input type="radio"/>	<input type="radio"/>

ANSWER:

Statements	Yes	No
The query performs a cross-partition query	<input type="radio"/>	<input checked="" type="radio"/>
The query uses an index	<input type="radio"/>	<input checked="" type="radio"/>
Recreating the container with the partition key set to /categoryId will improve the performance of the query	<input checked="" type="radio"/>	<input type="radio"/>

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>

#### QUESTION 51

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

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

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

NOTE: Each correct selection is worth one point.

A. In an Apache Spark pool in Azure Synapse, create a table that uses cosmos.olap as the data source.

B. Create a private endpoint connection to the account.

C. In an Azure Synapse Analytics serverless SQL pool, create a view that uses OPENROWSET and the CosmosDB provider.

D. Enable Azure Synapse Link for the account and Analytical store on the container.

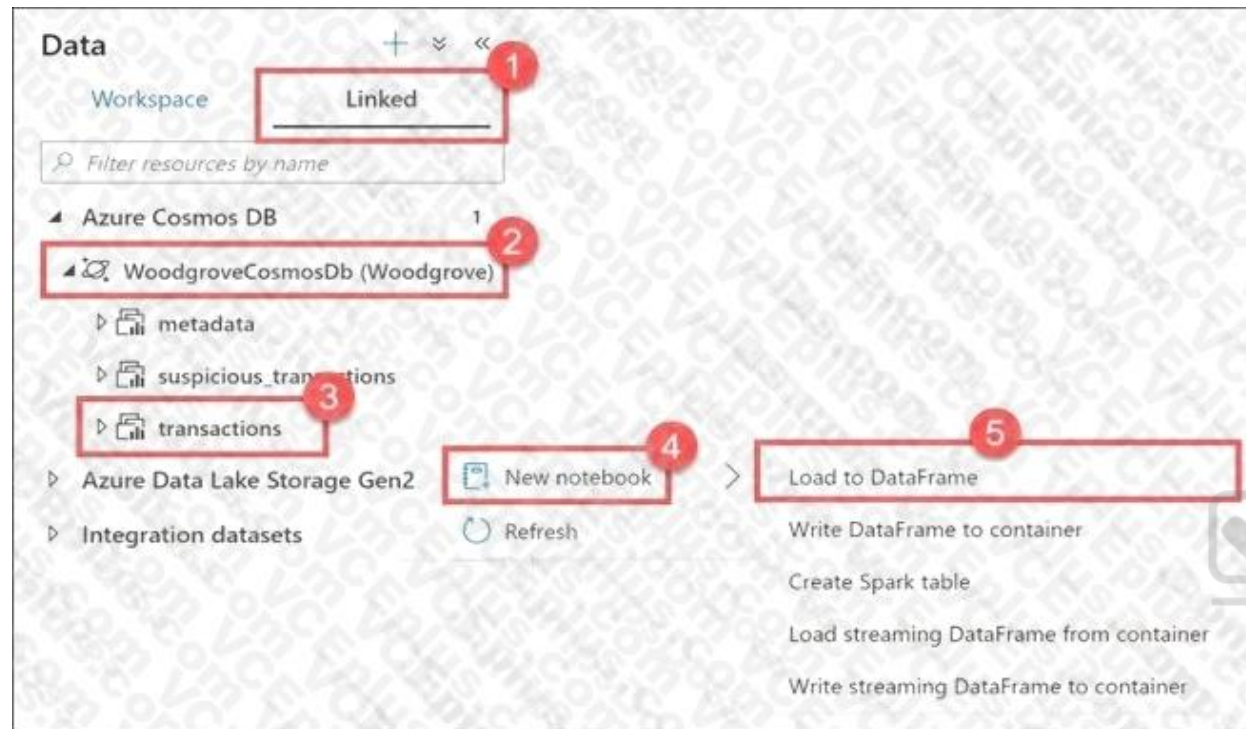
E. In an Apache Spark pool in Azure Synapse, create a table that uses cosmos.oltp as the data source.

ANSWER: A D

Explanation:

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.



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