

**100%** Money Back  
**Guarantee**

**Vendor:**Microsoft

**Exam Code:**DP-420

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

**Version:**Demo

## QUESTION 1

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

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

Which path will be indexed?

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

Correct Answer: A

The indexing policy has an includedPaths array that contains only one path:

/product/brand/. This means that only the properties under /product/brand will be indexed.

The symbol indicates that only scalar values will be indexed, not arrays or objects<sup>1</sup>.

The excludedPaths array contains a single path: /\* .

This means that all other properties will be excluded from indexing.

The \* symbol indicates a wildcard that matches any property name1.

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

---

## QUESTION 2

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

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

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

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

Does this meet the goal?

A. Yes

B. No

Correct Answer: B

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.

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

---

## QUESTION 3

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:

1.  
For container1 you must resolve conflicts by using the highest value for policyYear.
2.  
For container2 you must resolve conflicts by accepting the distance closest to latitude: 40.730610 and longitude: -73.935242.

3.  
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:



Correct Answer:

## Configurations

An application that reads from the conflicts feed

## Answer Area

Container1:

Last Write Wins (default) mode

Container2:

Merge Procedures (custom) mode

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 time-synchronization 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/en-us/azure/cosmos-db/sql/how-to-manage-conflicts>

---

## QUESTION 4

You have an Azure Cosmos DB for NoSQL account named account1 that supports an application named App1. App1 uses the consistent prefix consistency level.

You configure account1 to use a dedicated gateway and integrated cache.

You need to ensure that App1 can use the integrated cache.

Which two actions should you perform for APP1? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Change the connection mode to direct
- B. Change the account endpoint to <https://account1.sqlx.cosmos.azure.com>.
- C. Change the consistency level of requests to strong.

D. Change the consistency level of requests to session.

E. Change the account endpoint to <https://account1.documents.azure.com>

Correct Answer: BD

the Azure Cosmos DB integrated cache is an in-memory cache that is built-in to the Azure Cosmos DB dedicated gateway. The dedicated gateway is a front-end compute that stores cached data and routes requests to the backend database.

You can choose from a variety of dedicated gateway sizes based on the number of cores and memory needed for your workload<sup>1</sup>. The integrated cache can reduce the RU consumption and latency of read operations by serving them from

the cache instead of the backend containers<sup>2</sup>.

For your scenario, to ensure that App1 can use the integrated cache, you should perform these two actions:

Change the account endpoint to <https://account1.sqlx.cosmos.azure.com>. This is the dedicated gateway endpoint that you need to use to connect to your Azure Cosmos DB account and leverage the integrated cache. The standard gateway

endpoint (<https://account1.documents.azure.com>) will not use the integrated cache<sup>2</sup>.

Change the consistency level of requests to session. This is the highest consistency level that is supported by the integrated cache. If you use a higher consistency level (such as strong or bounded staleness), your requests will bypass the

integrated cache and go directly to the backend containers

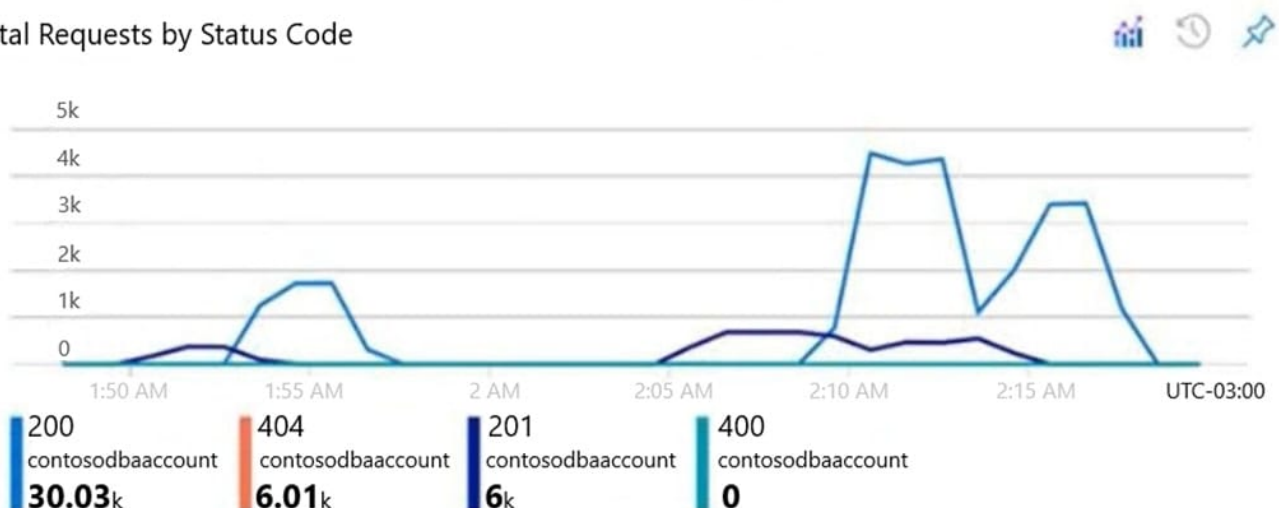
## QUESTION 5

### 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.

Total Requests by Status Code



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	

Correct 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	

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 6

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

Correct Answer: A

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 7

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
Bookauthorlnk	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 Bookauthorlnk 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 Bookauthorlnk.

Correct Answer: A

Store multiple entity types in the same container.

## QUESTION 8

You have a database named db1 in an Azure Cosmos DB Core (SQL) API account.

You have a third-party application that is exposed through a REST API.

You need to migrate data from the application to a container in db1 on a weekly basis.

What should you use?

- A. Database Migration Assistant
- B. Azure Data Factory
- C. Azure Migrate

Correct Answer: B

You can use Copy Activity in Azure Data Factory to copy data from and to Azure Cosmos DB (SQL API).

The Azure Cosmos DB (SQL API) connector is supported for the following activities:

Copy activity with supported source/sink matrix

Mapping data flow

Lookup activity

Incorrect:

Not A: Azure Migrate provides a centralized hub to assess and migrate on-premises servers, infrastructure, applications, and data to Azure. It assesses on-premises databases and migrate them to Azure SQL Database or to SQL Managed Instance.

Not C: Data Migration Assistant (DMA) enables you to upgrade to a modern data platform by detecting compatibility issues that can impact database functionality on your new version of SQL Server. It recommends performance and reliability

improvements for your target environment.

Reference:

<https://docs.microsoft.com/en-us/azure/data-factory/connector-azure-cosmos-db>

---

## QUESTION 9

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

Data update volumes are unpredictable.

You need to process the change feed of the container by using a web app that has multiple instances. The change feed will be processed by using the change feed processor from the Azure Cosmos DB SDK. The multiple instances must share the workload.

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

NOTE: Each correct selection is worth one point.

- A. Configure the same processor name for all the instances.
- B. Configure a different processor name for each instance.
- C. Configure a different lease container configuration for each instance.
- D. Configure the same instance name for all the instances. 13
- E. Configure a different instance name for each instance.
- F. Configure the same lease container configuration for all the instances.

Correct Answer: AEF

---

### QUESTION 10

You have the following query.

```
SELECT * FROM ? WHERE c.sensor = "TEMP1" AND c.value = 1619146031231
```

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

What should you recommend?

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

Correct Answer: A

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 >, Log Analytics logs– This alert triggers when the value of a specified property in the results of a Log Analytics query crosses a threshold you assign. For example, you can write a Log Analytics query to monitor if the storage for a logical

partition key is reaching the 20 GB logical partition key storage limit in Azure Cosmos DB.

Incorrect:

Metrics - The alert triggers when the value of a specified metric crosses a threshold you assign. For example, when the total request units consumed exceed 1000 RU/s. This alert is triggered both when the condition is first met and then

afterwards when that condition is no longer being met.

Activity log events – This alert triggers when a certain event occurs. For example, when the keys of your Azure Cosmos DB account are accessed or refreshed.

Reference:

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

---

## QUESTION 12

### 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	

Correct Answer:

## Answer Area

RequestOptions property to set:

	▼
AccessCondition	
ConsistencyLevel	
SessionToken	

Document property that will be compared:

	▼
_etag	
_id	
_rid	

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>