



DUMPSTODAY

Microsoft

DP-420 Exam

Azure Cosmos DB Developer Specialty

Questions & Answers

(Demo Version - Limited Content)

Thank you for Downloading DP-420 exam PDF Demo

Get Full File:

<https://dumpstoday.com/dp-420-dumps/>

WWW.DUMPSTODAY.COM

Question: 1

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>

Question: 2

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.

Partition key	<div style="border: 1px solid black; padding: 5px;"> <div style="border-bottom: 1px solid black; height: 20px; margin-bottom: 5px;"></div> <div style="border-bottom: 1px solid black; padding: 2px;">companyId</div> <div style="border-bottom: 1px solid black; padding: 2px;">companyId+emailAddress</div> <div style="border-bottom: 1px solid black; padding: 2px;">emailAddress</div> <div style="padding: 2px;">employeeId</div> </div>
Unique key	<div style="border: 1px solid black; padding: 5px;"> <div style="border-bottom: 1px solid black; height: 20px; margin-bottom: 5px;"></div> <div style="border-bottom: 1px solid black; padding: 2px;">companyId</div> <div style="border-bottom: 1px solid black; padding: 2px;">emailAddress</div> <div style="padding: 2px;">employeeId</div> </div>

Correct Answer:

Partition key	<input type="text"/>	▼
	companyId	
	companyId+emailAddress	
	emailAddress	
	employeeId	
Unique key	<input type="text"/>	▼
	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: 3

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:

Configure MaxItemCount in QueryRequestOptions	<input type="button" value=">"/> <input type="button" value="<"/>
Run the query and provide a continuation token	
Configure MaxBufferedItemCount in QueryRequestOptions	
Append the results to a variable	
Run the query and increment MaxItemCount	

Correct Answer:

The screenshot shows a query editor interface with two main sections: 'Actions' and 'Answer Area'. The 'Actions' section contains three steps: 1. 'Configure MaxItemLimit in QueryRequestOptions', 2. 'Configure MaxBufferedItemCount in QueryRequestOptions', and 3. 'Run the query and increment MaxItemLimit'. The 'Answer Area' contains three steps: 1. 'Configure MaxItemLimit in QueryRequestOptions', 2. 'Run the query and provide a continuation token', and 3. 'Append the results to a variable'. A right arrow button is positioned between the first and second actions, and a left arrow button is positioned between the second and third actions.

Explanation:

Step 1: Configure the MaxItemLimit in QueryRequestOptions

You can specify the maximum number of items returned by a query by setting the MaxItemLimit. The MaxItemLimit 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: 4

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": [
    {
```

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

Answer Area

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

Answer Area

Statements	Yes	No
The /isActivelyEnrolled property is included in the index	<input checked="" type="radio"/>	<input type="radio"/>
The /firsrtname 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"/>

Correct Answer:

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>

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

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

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.

Answer: A

Explanation:

Store multiple entity types in the same container.

Thank You for trying DP-420 PDF Demo

<https://dumpstoday.com/dp-420-dumps/>

Start Your DP-420 Preparation

[Limited Time Offer] Use Coupon "**SAVE20**" for extra 20% discount the purchase of PDF file. Test your DP-420 preparation with actual exam questions