



# DP-420<sup>Q&As</sup>

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





**Pass Microsoft DP-420 Exam with 100% Guarantee**

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.geekcert.com/dp-420.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





## QUESTION 1

### 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:

### Actions

Configure `MaxItemCount` in `QueryRequestOptions`

Run the query and provide a continuation token

Configure `MaxBufferedItemCount` in `QueryRequestOptions`

Append the results to a variable

Run the query and increment `MaxItemCount`

### Answer Area

Correct Answer:



## Actions

Configure MaxBufferedItemCount in QueryRequestOptions
Run the query and increment MaxItemCount

## Answer Area

Configure MaxItemCount in QueryRequestOptions
Run the query and provide a continuation token
Append the results to a variable

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-liveExplanation>:

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 2

### 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:



### Answer Area

Statements	Yes	No
<p>If you run the following query, the query will run as a cross-partition query</p> <pre>SELECT * FROM c where c.orderDate = "2019-05-03"</pre>	<input type="radio"/>	<input type="radio"/>
<p>If you run the following query, the query will run as a cross-partition query</p> <pre>SELECT * FROM c where c.customerId = "5425"</pre>	<input type="radio"/>	<input type="radio"/>
<p>If you run the following query, the query will run as a cross-partition query</p> <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"/>

Correct Answer:

### Answer Area

Statements	Yes	No
<p>If you run the following query, the query will run as a cross-partition query</p> <pre>SELECT * FROM c where c.orderDate = "2019-05-03"</pre>	<input checked="" type="radio"/>	<input type="radio"/>
<p>If you run the following query, the query will run as a cross-partition query</p> <pre>SELECT * FROM c where c.customerId = "5425"</pre>	<input checked="" type="radio"/>	<input type="radio"/>
<p>If you run the following query, the query will run as a cross-partition query</p> <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"/>

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 3

You need to create a database in an Azure Cosmos DB Core (SQL) API account.

The database will contain three containers named coll1, coll2, and coll3. The coll1 container will have unpredictable read and write volumes.

The coll2 and coll3 containers will have predictable read and write volumes.

The expected maximum throughput for coll1 and coll2 is 50,000 request units per second (RU/s) each.

How should you provision the collection while minimizing costs?

- A. Create a serverless account.
- B. Create a provisioned throughput account. Set the throughput for coll1 to Autoscale. Set the throughput for coll2 and coll3 to Manual.
- C. Create a provisioned throughput account. Set the throughput for coll1 to Manual. Set the throughput for coll2 and coll3 to Autoscale.

Correct Answer: B

Azure Cosmos DB offers two different capacity modes: provisioned throughput and serverless<sup>1</sup>. Provisioned throughput mode allows you to configure a certain amount of throughput (expressed in Request Units per second or RU/s) that is

provisioned on your databases and containers. You get billed for the amount of throughput you've provisioned, regardless of how many RUs were consumed<sup>1</sup>. Serverless mode allows you to run your database operations without having to

configure any previously provisioned capacity. You get billed for the number of RUs that were consumed by your database operations and the storage consumed by your data<sup>1</sup>. To create a database that minimizes costs, you should consider

the following factors:

The read and write volumes of your containers

The predictability and variability of your traffic  
The latency and throughput requirements of your application  
The geo-distribution and availability needs of your data  
Based on these factors, one possible option that you could choose is B. Create

a provisioned throughput account. Set the throughput for coll1 to Autoscale. Set the throughput for coll2 and coll3 to Manual.

This option has the following advantages:

It allows you to handle unpredictable read and write volumes for coll1 by using Autoscale, which automatically adjusts the provisioned throughput based on the current load<sup>1</sup>. It allows you to handle predictable read and write volumes for coll2



and coll3 by using Manual, which lets you specify a fixed amount of provisioned throughput that meets your performance needs1.

It allows you to optimize your costs by paying only for the throughput you need for each container1.

It allows you to enable geo-distribution for your account if you need to replicate your data across multiple regions1.

This option also has some limitations, such as:

It may not be suitable for scenarios where all containers have intermittent or bursty traffic that is hard to forecast or has a low average-to-peak ratio1. It may not be optimal for scenarios where all containers have low or sporadic traffic that

does not justify provisioned capacity1.

It may not support availability zones or multi-master replication for your account1. Depending on your specific use case and requirements, you may need to choose a different option. For example, you could use a serverless account if all

containers have low or sporadic traffic that does not require predictable performance or geo-distribution1 . Alternatively, you could use a provisioned throughput account with Manual for all containers if all containers have stable and consistent

traffic that requires predictable performance or geo-distribution1.

---

#### QUESTION 4

##### 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.

1.

id

2.

deviceid

3.

timestamp

4.

ownerid

5.

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:

### Answer 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"/>

Correct Answer:

### Answer Area

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"/>

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 5

### HOTSPOT

You have an Azure Cosmos DB Core (SQL) API account named account1 that has the disableKeyBasedMetadataWriteAccessproperty 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	

Correct 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	

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/en-us/rest/api/resources/>

[DP-420 Study Guide](#)

[DP-420 Exam Questions](#)

[DP-420 Braindumps](#)