



Up-to-date Practice Test with Latest Questions and Answers covering latest syllabus and topics of the exam. Makes you ready to face actual exam.



DP-420 Practice Questions
DP-420 Practice Test
DP-420 Practice Exam
DP-420 Exam Questions
DP-420 Study Guide



killexams.com

Microsoft

DP-420

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

ORDER FULL VERSION



<https://killexams.com/pass4sure/exam-detail/DP-420>

Question: 20

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

Settings

Indexing Policy

Time to Live

- Off
- On (no default)
- On

Geospatial Configuration

- Geography
- Geometry

Partition key

/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: 21

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:

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

Question: 22

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

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

You have a container named container1 in an Azure Cosmos DB for NoSQL 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 to copy data to another Azure Cosmos DB for NoSQL container.

Does this meet the goal?

- A. Yes
- B. No

Answer: B

Question: 23

You have a container in an Azure Cosmos DB for NoSQL account. The container stores data about families. Data about parents, children, and pets are stored as separate documents.

Each document contains the address of each family. Members of the same family share the same partition key named family Id

You need to update the address for each member of the same family that share the same address.

The solution must meet the following requirements:

- â€¢ Be atomic consistent isolated, and durable (ACID).
- â€¢ Provide the lowest latency.

What should you do?

- A. Update the document of each family member by using a transactional batch operation.
- B. Update the document of each family member separately by using a patch operation.
- C. Update the document of each family member separately and set the consistency level to strong.

Answer: A

Question: 24

HOTSPOT

You have an Azure Cosmos DB for NoSQL account named account that has the disablekey Basedletadatwrite Access property enabled.

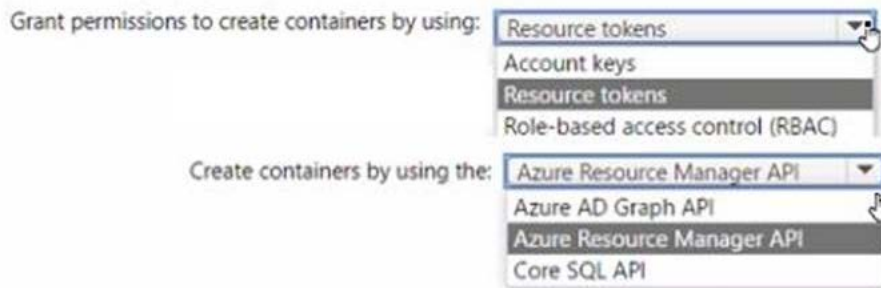
You are developing an app named App1 that will be used by a user named DevUser1 to create containers in account1. DevUser1 has a non-privileged user account in the Azure AD tenant.

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.

Answer Area



- wrong

Answer: A

Question: 25

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

What should you do to resolve the issue?

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

Answer: C

Question: 26

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.

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"/>

- wrong

Answer: C

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: 27

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: D

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: 28

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.

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"/>

- wrong

Answer: D

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.

Question: 29

You are building an application that will store data in an Azure Cosmos DB for NoSQL account. The account uses the session default consistency level. The account is used by five other applications. The account has a single read-write region and 10 additional read regions.

Approximately 20 percent of the items stored in the account are updated hourly.

Several users will access the new application from multiple devices.

You need to ensure that the users see the same item values consistently when they browse from the different devices. The solution must not affect the other applications.

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

- A. Use implicit session management when performing read requests.
- B. Provide a stored session token when performing read requests.
- C. Associate a session token to the user account.
- D. Set the default consistency level to eventual.
- E. Associate a session token to the device.

Answer: A,B,C

Question: 30

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.

RequestOptions property to set:

	▼
AccessCondition	
ConsistencyLevel	
SessionToken	

Document property that will be compared:

	▼
_etag	
_id	
_rid	

- wrong

Answer: A,B,C

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.

Question: 31

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: 32

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: 33

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.

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

Question: 34

HOTSPOT

You have an Azure Cosmos DB for NoSQL account.

You plan to create a container named container1. The container1 container will store items that include two properties named nm and age.

The most commonly executed queries will query container1 for a specific name.

The following is a sample of the query.

```
SELECT *  
FROM c  
WHERE c.name = 'ben smith'  
ORDER BY c.name DESC, c.age ASC
```

You need to define an opt-in Indexing policy for container1.

The solution must meet the following requirements:

- â€¢ Minimize the number of request units consumed by the queries.
- â€¢ Ensure that the _etag property is excluded from indexing.

How should you define the indexing policy? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

Answer Area

```
{
  "automatic":true,
  "indexingMode":"Consistent",
  "includedPaths":[],
  "excludedPaths":
  [{"path":"/^*"}],
  "compositeIndexes":[
    {
      "path":"/name/*",
      "order":"asc"
    },
    {
      "path":"/age",
      "order":"descending"
    }
  ]
}
```

- wrong

Answer: D

Question: 35

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 email Address value.

You need to ensure that the email Address 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

	▼
companyId	
companyId+emailAddress	
emailAddress	
employeeId	

Unique key

	▼
companyId	
emailAddress	
employeeId	

- wrong

Answer: D

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

Killexams.com is a leading online platform specializing in high-quality certification exam preparation. Offering a robust suite of tools, including Exam Questions, practice tests, and advanced test engines, Killexams.com empowers candidates to excel in their certification exams. Discover the key features that make Killexams.com the go-to choice for exam success.



Practice Exam Questions Based on Current Exam Objectives

Killexams.com provides practice exam questions aligned with the latest official exam objectives and latest syllabus. Our content is reviewed and updated regularly to reflect recent changes announced by certification vendors. By studying these practice questions, candidates will cover the structure, difficulty level, and topics of the actual exam, helping them prepare more effectively and efficiently.

Comprehensive Practice Exams (PDF Format)

Killexams.com offers multiple-choice questions (MCQs) in easy-to-read PDF format, covering all major domains of the exam. Each PDF contains a structured collection of practice questions and verified answers designed to support focused study. These MCQs help candidates reinforce key concepts, identify knowledge gaps, and improve exam readiness through consistent practice.

Realistic Practice Tests (Online Test Engine & Desktop Test Engine)

To support hands-on preparation, Killexams.com provides practice tests through both an Online Test Engine and a Desktop Test Engine. These tools are designed to simulate a real exam environment, allowing candidates to practice under exam-like conditions, with latest syllabus and topics of the exam. Performance tracking, test history, and result analysis help users evaluate their progress and focus on areas that need improvement.

Risk-Free Purchase Policy

Killexams.com follows a transparent and customer-friendly purchase policy. If users are not satisfied with the study materials, they may request assistance or a refund in accordance with our published terms and conditions. This policy reflects our commitment to customer satisfaction, fairness, and confidence in our preparation resources.

Regularly Updated Content

Our practice question bank is reviewed and updated on an ongoing basis to stay aligned with the latest exam outlines and vendor updates. This ensures candidates are studying up-to-date, relevant material, and preparing with content that reflects current exam expectations, helping them stay confident and well-prepared.