



DP-420^{Q&As}

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.passapply.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

You have an Azure Cosmos DB for NoSQL account.

You need to create an Azure Monitor query that lists recent modifications to the regional failover policy.

Which Azure Monitor table should you query?

- A. CDBPartitionKeyStatistics
- B. CDBQueryRunTimeStatistics
- C. CDBDataPlaneRequests
- D. CDBControlPlaneRequests

Correct Answer: D

QUESTION 2

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 pWHERE p.accountnumber = \'accountnumber\'"`
- B. `cmd = "SELECT * FROM Persons pWHERE p.accountnumber = LIKE @accountnumber"`
- C. `cmd = "SELECT * FROM Persons pWHERE p.accountnumber = @accountnumber"`
- D. `cmd = "SELECT * FROM Persons pWHERE p.accountnumber = \' + accountnumber + \'"`

Correct Answer: C

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 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:

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 4

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:

Answer 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"/> |



Correct Answer:

Answer Area

| 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"/> |

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/cosmos-db/optimize-cost-regions>

QUESTION 5

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 > ,