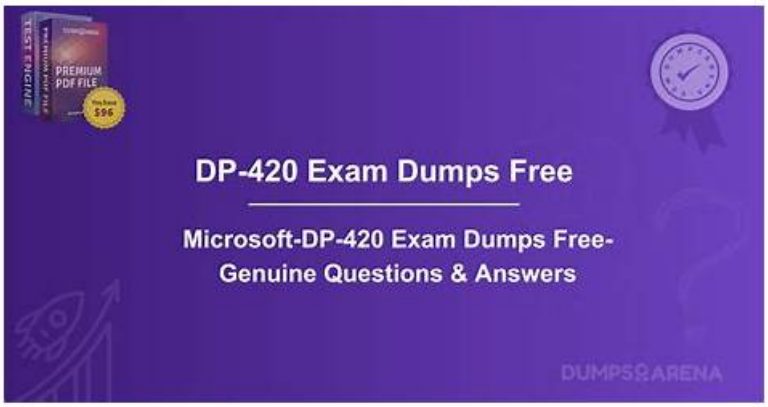
DP-420 Braindumps Downloads - Dumps DP-420 Free



DOWNLOAD the newest Actual4dump DP-420 PDF dumps from Cloud Storage for free: https://drive.google.com/open?id=1ikQ_w_M05eseAPdKdwl9S2Y4uum3Tnjv

Through our investigation and analysis of the real problem over the years, our DP-420 learning materials can accurately predict the annual DP-420 exams. In the actual exam process, users will encounter almost half of the problem is similar in our products. Even if the syllabus is changing every year, the DP-420 Study Materials' experts still have the ability to master propositional trends. Believe that such a high hit rate can better help users in the review process to build confidence, and finally help users through the qualification examination to obtain a certificate.

The DP-420 Exam is intended for solution architects, developers, and database administrators who have experience working with Azure Cosmos DB and want to showcase their skills in designing and implementing cloud-native applications using this technology. Designing and Implementing Cloud-Native Applications Using Microsoft Azure Cosmos DB certification exam tests the candidate's ability to design and implement containers, partitions, and indexes in Azure Cosmos DB to optimize performance, scalability, and cost.

>> DP-420 Braindumps Downloads <<

Dumps DP-420 Free, DP-420 100% Correct Answers

Our company employs a professional service team which traces and records the popular trend among the industry and the latest update of the knowledge about the DP-420 exam reference. We give priority to keeping pace with the times and providing the advanced views to the clients. We keep a close watch at the most advanced social views about the knowledge of the test DP-420 Certification. Our experts will renovate the test bank with the latest DP-420 exam practice question and compile the latest knowledge and information into the DP-420 exam questions and answers.

The DP-420 certification exam covers a range of topics related to the design and implementation of cloud-native applications using Azure Cosmos DB. These topics include data modeling, partitioning, indexing, querying, and managing data at scale. DP-420 Exam also covers topics related to the use of Azure Cosmos DB in cloud-native architectures, including the use of Azure Functions, Azure Event Grid, and Azure Service Bus.

Microsoft Designing and Implementing Cloud-Native Applications Using Microsoft Azure Cosmos DB Sample Questions (Q39-Q44):

NEW QUESTION # 39
You have a database in an Azure Cosmos DB for NoSQL account that is configured for multi-region writes.
You need to use the Azure Cosmos DB SDK to implement the conflict resolution policy for a container. The solution must ensure that any conflict sent to the conflict feed.
Solution: You set ConfilictResolutionMode to Custom. You Set ResolutionProcedures to a custom stored procedure. You configure the custom stored procedure to use the isTomstone parameter to resolve conflict.
Does this meet the goal?

- A. No
- B. Yes

Answer: B

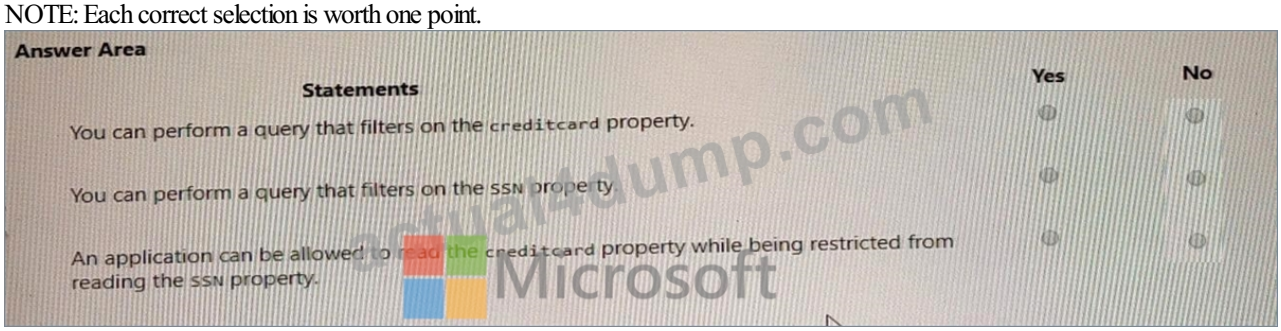
Explanation:
Explanation
The solution is incorrect because there is no "isTom" parameter in the Azure Cosmos DB SDK. The correct parameter is "isTombstone".

NEW QUESTION # 40
You have a container named container1 in an Azure Cosmos DB for NoSQL account named account1.
You configure container1 to use Always Encrypted by using an encryption policy as shown in the C# and the Java exhibits. (Click the C# tab to view the encryption policy in C#. Click the Java tab to see the encryption policy in Java.)

```
var path1 = new ClientEncryptionIncludedPath
{
    Path = "/creditcard",
    ClientEncryptionKeyId = "encryptionkey",
    EncryptionType = EncryptionType.Randomized.ToString(),
    EncryptionAlgorithm = DataEncryptionKeyAlgorithm.AEAD_AES_256_CBC_HMAC_SHA256.ToString()
};
var path2 = new ClientEncryptionIncludedPath
{
    Path = "/SSN",
    ClientEncryptionKeyId = "encryptionkey",
    EncryptionType = EncryptionType.Deterministic.ToString(),
    EncryptionAlgorithm = DataEncryptionKeyAlgorithm.AEAD_AES_256_CBC_HMAC_SHA256.ToString()
};

await database.DefineContainer("container1", "/partitionkey")
    .WithClientEncryptionPolicy()
    .WithIncludedPath(path1)
    .WithIncludedPath(path2)
    .Attach()
    .CreateAsync();
```

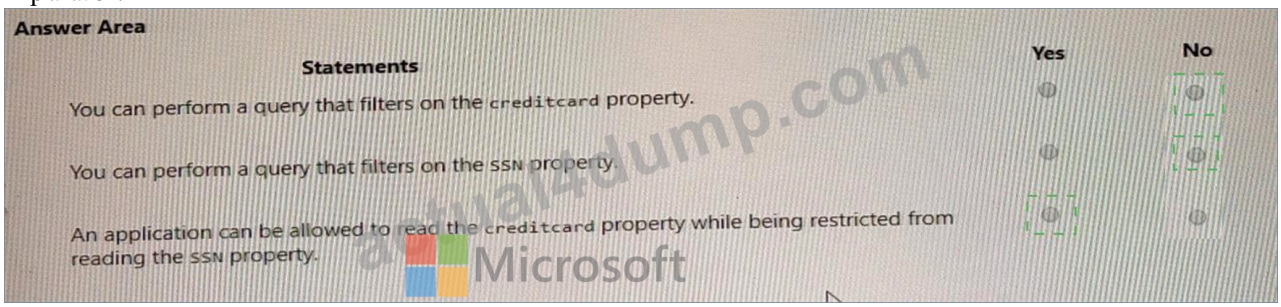
```
ClientEncryptionIncludedPath path1 = new ClientEncryptionIncludedPath();
path1.path = "/creditcard";
path1.clientEncryptionKeyId = "encryptionkey";
path1.encryptionType = CosmosEncryptionType.RANDOMIZED;
path1.encryptionAlgorithm = CosmosEncryptionAlgorithm.AEAES_256_CBC_HMAC_SHA_256;

ClientEncryptionIncludedPath path2 = new ClientEncryptionIncludedPath();
path2.path = "/SSN";
path2.clientEncryptionKeyId = "encryptionkey";
path2.encryptionType = CosmosEncryptionType.DETERMINISTIC;
path2.encryptionAlgorithm = CosmosEncryptionAlgorithm.AEAES_256_CBC_HMAC_SHA_256;

entEncryptionIncludedPath> paths = new ArrayList<>();
paths.add(path1);
paths.add(path2);

CosmosContainerProperties containerProperties =
    new CosmosContainerProperties("container1", "/partitionkey");
containerProperties.setClientEncryptionPolicy(new ClientEncryptionPolicy(paths));
database.createEncryptionContainerAsync(containerProperties);
```

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:

Explanation:



Explanation

According to the Azure Cosmos DB documentation1, Always Encrypted is a feature designed to protect sensitive data, such as credit card numbers or national identification numbers, stored in Azure Cosmos DB.

Always Encrypted allows clients to encrypt sensitive data inside client applications and never reveal the encryption keys to the database.

To use Always Encrypted, you need to define an encryption policy for each container that specifies which properties should be encrypted and which data encryption keys (DEK) should be used. The DEKs are stored in Azure Cosmos DB and are wrapped by customer-managed keys (CMK) that are stored in Azure Key Vault.

Based on the encryption policy shown in the exhibits, the creditcard property is encrypted with a DEK named dek1, and the SSN property is encrypted with a DEK named dek2. Both DEKs are wrapped by a CMK named cmk1.

To answer your statements:

You can perform a query that filters on the creditcard property = No. This is because the creditcard property is encrypted and cannot be used for filtering or sorting operations1.

You can perform a query that filters on the SSN property = No. This is also because the SSN property is encrypted and cannot be used for filtering or sorting operations1.

An application can be allowed to read the creditcard property while being restricted from reading the SSN property = Yes. This is possible by using different CMKs to wrap different DEKs and applying access policies on the CMKs in Azure Key Vault. For example, if you use cmk2 to wrap dek2 instead of cmk1, you can grant an application access to cmk1 but not cmk2, which means it can read the creditcard property but not the SSN property2.

NEW QUESTION # 41

You have an Azure Cosmos DB Core (SQL) API account named storage1 that uses provisioned throughput capacity mode.

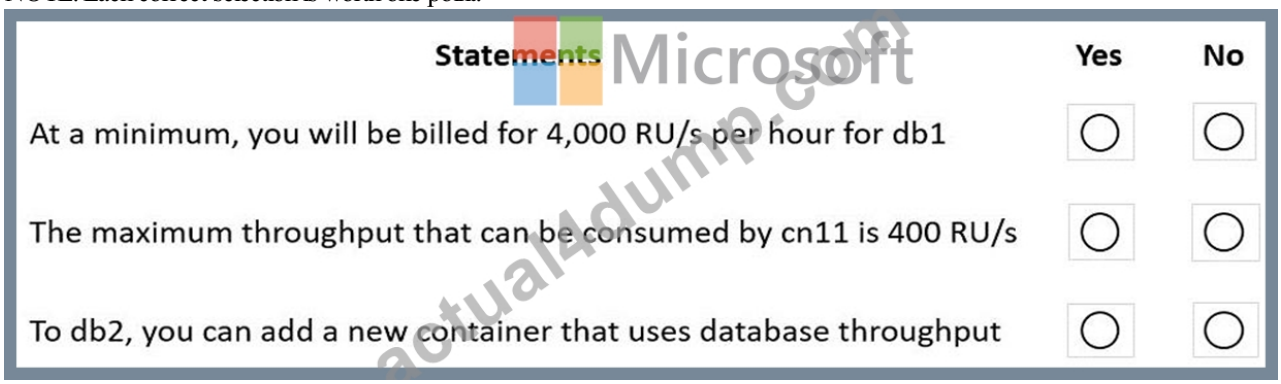
The storage1 account contains the databases shown in the following table.

Name	Throughput	Max request units per second (RU/s)	Geo-redundancy	Multi-region writes	Number of regions
db1	Autoscale	5,000	Disabled	Disabled	1
db2	Autoscale	8,000	Enabled	Enabled	3

The databases contain the containers shown in the following table.

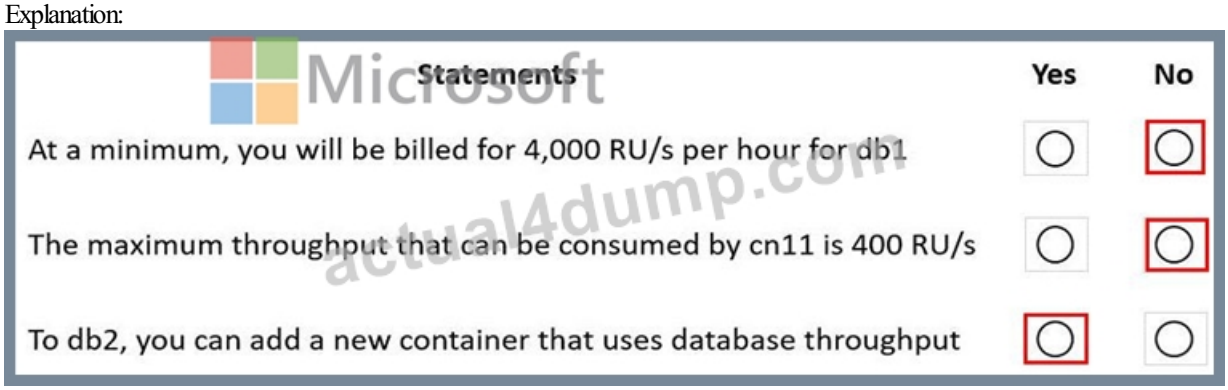
Name	Database	Throughput
cn01	db1	Container - autoscale maximum RU/s of 10,000
cn02	db1	Database
cn03	db1	Database
cn04	db1	Database
cn05	db1	Database
cn11	db2	Database
cn12	db2	Database
cn13	db2	Database
cn14	db2	Database
cn15	db2	Database
cn16	db2	Database
cn17	db2	Database
cn18	db2	Database

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:

Explanation:



Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/plan-manage-costs
https://azure.microsoft.com/en-us/pricing/details/cosmos-db/

NEW QUESTION # 42
You have an application that queries an Azure Cosmos 06 for NoSQL account.
You discover that the following two queries run frequently,

You need to minimize the request units (RUs) consumed by reads and writes. What should you create?

- A. a composite index for (name ASC, time stamp DESC)

- B. a composite index for (name ASC time stamp ASC) and a composite index for (name, time stamp disc)
- C. a composite index for (name DESC, time stamp ASC)
- D. a composite index for (name ASC, time stamp ASC)

Answer: D

NEW QUESTION # 43
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.

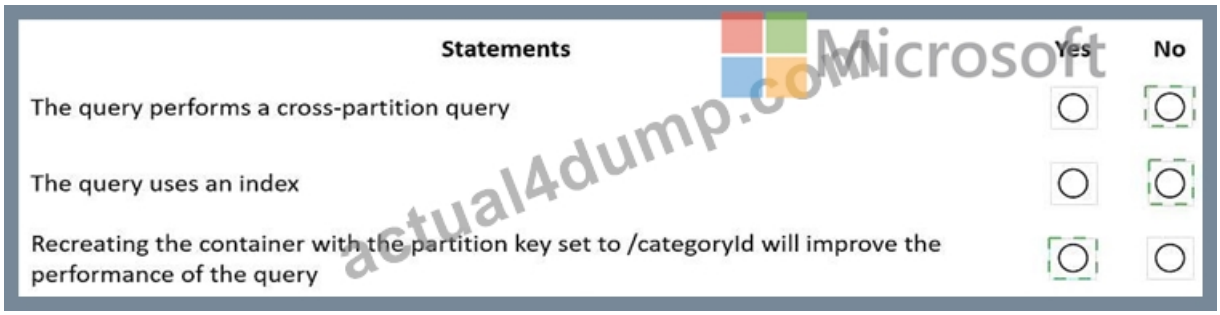
```
Retrieved Document Count                    :          45,654
Retrieved Document Size                     :     543,765,234 bytes
Output Document Count                       :              12
Output Document Size                        :             451 bytes
Index Utilization                           :            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.

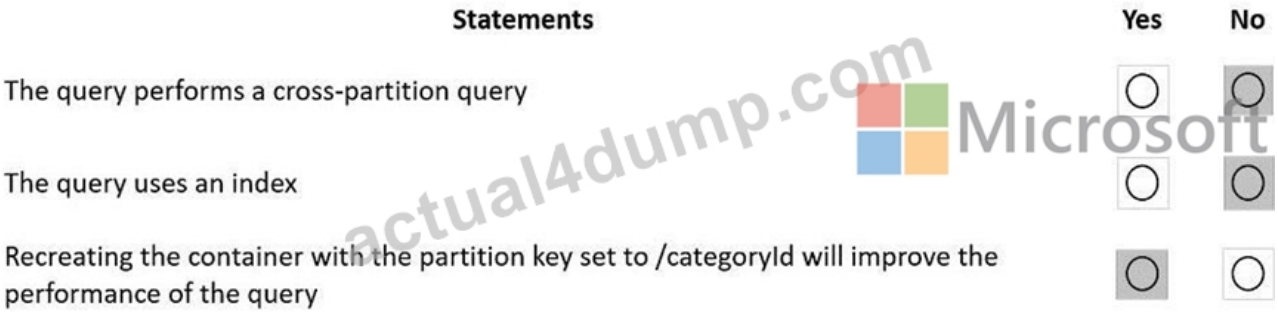
Statements	Yes	No
The query performs a cross-partition query	○	○
The query uses an index	○	○
Recreating the container with the partition key set to /categoryId will improve the performance of the query	○	○

Answer:

Explanation:



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.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-query-container

NEW QUESTION # 44
.....

- www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, 61921c.com, building.lv, automastery.in, daotao.wisebusiness.edu.vn, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, masteringdigitalskills.com, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes

P.S. Free & New DP-420 dumps are available on Google Drive shared by Actual4dump: https://drive.google.com/open?id=1ikQ_w_M05eseAPdKdwl9S2Y4uum3Tnjv