



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 need to create a data store for a directory of small and medium-sized businesses (SMBs). The data store must meet the following requirements:

1.
Store companies and the users employed by them. Each company will have less than 1,000 users.
2.
Some users have data that is greater than 2 KB.
3.
Associate each user to only one company.
4.
Provide the ability to browse by company.
5.
Provide the ability to browse the users by company.
6.
Whenever a company or user profile is selected, show a details page for the company and all the related users.
7.
Be optimized for reading data.

Which design should you implement to optimize the data store for reading data?

- A. In a directory container, create a document for each company and a document for each user. Use company ID as the partition key.
- B. In a company container, create a document for each company. Embed the users into company documents. Use the company ID as the partition key.
- C. Create a user container that uses the user ID as the partition key and a company container that container that uses the company ID as the partition key. Add the company ID to each user documents.
- D. In a user container, create a document for each user. Embed the company into each user document. Use the user ID as the partition key.

Correct Answer: B

Azure Cosmos DB is a multi-model database that supports various data models, such as documents, key-value, graph, and column-family³. The core content-model of Cosmos DB's database engine is based on atom-record-sequence

(ARS), which allows it to store and query different types of data in a flexible and efficient way³. To optimize the data store for reading data, you should consider the following factors:



The size and shape of your data

The frequency and complexity of your queries

The latency and throughput requirements of your application The trade-offs between storage efficiency and query performance

Based on these factors, one possible design that you could implement is B. In a company container, create a document for each company. Embed the users into company documents. Use the company ID as the partition key.

This design has the following advantages:

It stores companies and users as self-contained documents that can be easily retrieved by company ID¹.

It avoids storing redundant data or creating additional containers for users¹. It allows you to browse by company and browse the users by company with simple queries¹.

It shows a details page for the company and all the related users by fetching a single document¹.

It leverages the benefits of embedding data, such as reducing the number of requests, improving query performance, and simplifying data consistency².

This design also has some limitations, such as:

It may not be suitable for some users who have data that is greater than 2 KB, as it could exceed the maximum document size limit of 2 MB². It may not be optimal for scenarios where users need to be associated with more than one

company or queried independently from companies². It may not be scalable for companies that have more than 1,000 users, as it could result in hot partitions or throttling². Depending on your specific use case and requirements, you may

need to adjust this design or choose a different one. For example, you could use a hybrid data model that combines embedding and referencing data², or you could use a graph data model that expresses entities and relationships as vertices

and edges.

QUESTION 2

You have an Azure Cosmos DB Core (SQL) API account.

You configure the diagnostic settings to send all log information to a Log Analytics workspace.

You need to identify when the provisioned request units per second (RU/s) for resources within the account were modified.

You write the following query.

AzureDiagnostics

| where Category == "ControlPlaneRequests"

What should you include in the query?

A. | where OperationName startswith "AccountUpdateStart"



- B. | where OperationName startswith "SqlContainersDelete"
- C. | where OperationName startswith "MongoCollectionsThroughputUpdate"
- D. | where OperationName startswith "SqlContainersThroughputUpdate"

Correct Answer: A

The following are the operation names in diagnostic logs for different operations:

1.
RegionAddStart, RegionAddComplete
2.
RegionRemoveStart, RegionRemoveComplete
3.
AccountDeleteStart, AccountDeleteComplete
4.
RegionFailoverStart, RegionFailoverComplete
5.
AccountCreateStart, AccountCreateComplete
6.
AccountUpdateStart, AccountUpdateComplete
7.
VirtualNetworkDeleteStart, VirtualNetworkDeleteComplete
8.
DiagnosticLogUpdateStart, DiagnosticLogUpdateComplete

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/audit-control-plane-logs>

QUESTION 3

You plan to create an operational system that will store data in an Azure Cosmos DB or NoSQL account. You need to configure the account to meet the following requirements:

1.
Support Spar* queries.
- 2.



Support the analysis of data from the last six months.

3.

Only pay for analytical compute when running queries.

Which three actions should you perform?

Each correct answer presents part of the solution. NOTE Each correct selection is worth one point.

- A. Create an Azure Synapse linked service.
- B. Create a container and set the time to live to six months.
- C. Create a container and set the analytical property to six months.
- D. Create an Azure Synapse pipeline.
- E. Create an Azure Databricks notebook.
- F. Enable Azure Synapse Link for the account

Correct Answer: CEF

QUESTION 4

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.

Correct Answer: BC



QUESTION 5

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

Correct Answer: D

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>

[Latest DP-420 Dumps](#)

[DP-420 PDF Dumps](#)

[DP-420 VCE Dumps](#)