



# 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

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-family<sup>3</sup>. 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 way<sup>3</sup>. 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 ID<sup>1</sup>.

It avoids storing redundant data or creating additional containers for users<sup>1</sup>. It allows you to browse by company and browse the users by company with simple queries<sup>1</sup>.

It shows a details page for the company and all the related users by fetching a single document<sup>1</sup>.

It leverages the benefits of embedding data, such as reducing the number of requests, improving query performance, and simplifying data consistency<sup>2</sup>.

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 MB<sup>2</sup>. It may not be optimal for scenarios where users need to be associated with more than one

company or queried independently from companies<sup>2</sup>. It may not be scalable for companies that have more than 1,000 users, as it could result in hot partitions or throttling<sup>2</sup>. 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 data<sup>2</sup>, or you could use a graph data model that expresses entities and relationships as vertices

and edges.

---

## QUESTION 2

You need to provide a solution for the Azure Functions notifications following updates to con-product. The solution must meet the business requirements and the product catalog requirements.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Configure the trigger for each function to use a different leaseCollectionPrefix
- B. Configure the trigger for each function to use the same leaseCollectionName
- C. Configure the trigger for each function to use a different leaseCollectionName
- D. Configure the trigger for each function to use the same leaseCollectionPrefix

Correct Answer: AC



leaseCollectionPrefix: when set, the value is added as a prefix to the leases created in the Lease collection for this Function. Using a prefix allows two separate Azure Functions to share the same Lease collection by using different prefixes.

Scenario: Use Azure Functions to send notifications about product updates to different recipients.

Trigger the execution of two Azure functions following every update to any document in the con-product container.

Reference:

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

---

### QUESTION 3

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account.

You need to provide a user named User1 with the ability to insert items into container1 by using role-based access control (RBAC). The solution must use the principle of least privilege.

Which roles should you assign to User1?

- A. CosmosDB Operator only
- B. DocumentDB Account Contributor and Cosmos DB Built-in Data Contributor
- C. DocumentDB Account Contributor only
- D. Cosmos DB Built-in Data Contributor only

Correct Answer: A

Cosmos DB Operator: Can provision Azure Cosmos accounts, databases, and containers. Cannot access any data or use Data Explorer. Incorrect Answers:

B: DocumentDB Account Contributor can manage Azure Cosmos DB accounts. Azure Cosmos DB is formerly known as DocumentDB.

C: DocumentDB Account Contributor: Can manage Azure Cosmos DB accounts.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/role-based-access-control>

---

### QUESTION 4

You are implementing an Azure Data Factory data flow that will use an Azure Cosmos DB (SQL API) sink to write a dataset. The data flow will use 2,000 Apache Spark partitions. You need to ensure that the ingestion from each Spark partition is balanced to optimize throughput.

Which sink setting should you configure?

- A. Throughput
- B. Write throughput budget



C. Batch size

D. Collection action

Correct Answer: C

Batch size: An integer that represents how many objects are being written to Cosmos DB collection in each batch. Usually, starting with the default batch size is sufficient. To further tune this value, note:

Cosmos DB limits single request's size to 2MB. The formula is "Request Size = Single Document Size \* Batch Size". If you hit error saying "Request size is too large", reduce the batch size value.

The larger the batch size, the better throughput the service can achieve, while make sure you allocate enough RUs to empower your workload.

Incorrect Answers:

A: Throughput: Set an optional value for the number of RUs you'd like to apply to your CosmosDB collection for each execution of this data flow. Minimum is 400.

B: Write throughput budget: An integer that represents the RUs you want to allocate for this Data Flow write operation, out of the total throughput allocated to the collection.

D: Collection action: Determines whether to recreate the destination collection prior to writing.

None: No action will be done to the collection. Recreate: The collection will get dropped and recreated

Reference: <https://docs.microsoft.com/en-us/azure/data-factory/connector-azure-cosmos-db>

## QUESTION 5

You have an Azure Cosmos DB for NoSQL account named account1 that has a single read-write region and one additional read region.

Account1 uses the strong default consistency level.

You have an application that uses the eventual consistency level when submitting requests to account1.

How will writes from the application be handled?

A. Writes will use the strong consistency level.

B. Azure Cosmos DB will reject writes from the application.

C. The write order is not guaranteed during replication.

D. Writes will use the eventual consistency level.

Correct Answer: A

This is because the write concern is mapped to the default consistency level configured on your Azure Cosmos DB account, which is strong in this case. Strong consistency ensures that every write operation is synchronously committed to every region associated with your Azure Cosmos DB account. The eventual consistency level that the application uses only applies to the read operations. Eventual consistency offers higher availability and better performance, but it does not guarantee the order or latency of the reads.



VCE & PDF

GeekCert.com

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

2024 Latest geekcert DP-420 PDF and VCE dumps Download

---

[DP-420 PDF Dumps](#)

[DP-420 Exam Questions](#)

[DP-420 Braindumps](#)