

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.leads4pass.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 configure multi-region writes for account1.

You need to ensure that App1 supports the new configuration for account1. The solution must meet the business requirements and the product catalog requirements.

What should you do?

- A. Set the default consistency level of account1 to bounded staleness.
- B. Create a private endpoint connection.
- C. Modify the connection policy of App1.
- D. Increase the number of request units per second (RU/s) allocated to the con-product and con-productVendor containers.

Correct Answer: D

App1 queries the con-product and con-productVendor containers.

Note: Request unit is a performance currency abstracting the system resources such as CPU, IOPS, and memory that are required to perform the database operations supported by Azure Cosmos DB.

Scenario:

Develop an app named App1 that will run from all locations and query the data in account1.

Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.

Whenever there are multiple solutions for a requirement, select the solution that provides the best performance, as long as there are no additional costs associated.

Incorrect Answers:

A:

Bounded staleness relates to writes. App1 only do reads.

Note: Bounded staleness is frequently chosen by globally distributed applications that expect low write latencies but require total global order guarantee. Bounded staleness is great for applications featuring group collaboration and sharing, stock ticker, publish-subscribe/queueing etc.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/consistency-levels>

QUESTION 2

You need to implement a solution to meet the product catalog requirements. What should you do to implement the conflict resolution policy.

- A. Remove frequently changed field from the index policy of the con-product container.

- B. Disable indexing on all fields in the index policy of the con-product container.
- C. Set the default consistency level for account1 to eventual.
- D. Create a new container and migrate the product catalog data to the new container.

Correct Answer: D

QUESTION 3

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:

1.

You set ConflictResolutionMode to Custom.

2.

You Set ResolutionProcedures to a custom stored procedure.

3.

You configure the custom stored procedure to use the conflictingItems parameter to resolve conflict.

Does this meet the goal?

A. Yes

B. No

Correct Answer: A

Setting ConflictResolutionMode to Custom and configuring a custom stored procedure with the "conflictingItems" parameter will allow you to implement a custom conflict resolution policy. This will ensure that any conflicts are sent to the conflict feed for resolution.

QUESTION 4

You are developing an application that will use an Azure Cosmos DB Core (SQL) API account as a data source. You need to create a report that displays the top five most ordered fruits as shown in the following table.

Name	Type	Orders
apple	fruit	1,000
orange	fruit	600
banana	fruit, exotic	400
plum	fruit	300
mango	fruit, exotic	200

A collection that contains aggregated data already exists. The following is a sample document:

```
{  
  "name": "apple",  
  "type": ["fruit", "exotic"],  
  "orders": 10000  
}
```

Which two queries can you use to retrieve data for the report? Each correct answer presents a complete solution.
NOTE: Each correct selection is worth one point.

- A.

```
SELECT TOP i.name, i.types, i.orders  
FROM items i  
WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit')  
ORDER BY i.orders,i.types
```
- B.

```
SELECT TOP i.name, i.types, i.orders  
FROM items i  
WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit')  
ORDER BY i.orders DESC
```
- C.

```
SELECT TOP i.name, i.types, i.orders  
FROM items i  
WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit')  
ORDER BY i.types DESC
```
- D.

```
SELECT TOP i.name, i.types, i.orders  
FROM items i  
WHERE ARRAY_CONTAINS(i.types, {name: 'fruit'})  
ORDER BY i.orders DESC
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: BD

ARRAY_CONTAINS returns a Boolean indicating whether the array contains the specified value. You can check for a partial or full match of an object by using a boolean expression within the command. Incorrect Answers:

A: Default sorting ordering is Ascending. Must use Descending order.

C: Order on Orders not on Type.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/sql-query-array-contains>

QUESTION 5

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.

[Latest DP-420 Dumps](#)

[DP-420 PDF Dumps](#)

[DP-420 Practice Test](#)