



# 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.pass4itsure.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

### HOTSPOT

You are creating a database in an Azure Cosmos DB Core (SQL) API account. The database will be used by an application that will provide users with the ability to share online posts. Users will also be able to submit comments on other

users\' posts.

You need to store the data shown in the following table.

Type	Description
Users	Information about a user who will use the application
Posts	Text of up to 1,000 characters that a user will share with other users
Comments	Text of up to 280 characters that users will submit as a comment on a post
Interests	Information about a user\'s interests

The application has the following characteristics:

1.

Users can submit an unlimited number of posts.

2.

The average number of posts submitted by a user will be more than 1,000.

3.

Posts can have an unlimited number of comments from different users.

4.

The average number of comments per post will be 100, but many posts will exceed 1,000 comments.

5.

Users will be limited to having a maximum of 20 interests.

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
If you embed the posts data into the users data instead of creating a separate document for each post, you will increase the write operation costs for new posts	<input type="radio"/>	<input type="radio"/>
If you embed the comments data into the posts data instead of creating a separate document for each comment you will increase the write operation costs for new comments	<input type="radio"/>	<input type="radio"/>
If you embed the interests data into the users data instead of creating a separate document for each interest, you will increase the read operation costs for displaying the users and their associated interests	<input type="radio"/>	<input type="radio"/>

Correct Answer:

### Answer Area

Statements	Yes	No
If you embed the posts data into the users data instead of creating a separate document for each post, you will increase the write operation costs for new posts	<input checked="" type="radio"/>	<input type="radio"/>
If you embed the comments data into the posts data instead of creating a separate document for each comment you will increase the write operation costs for new comments	<input checked="" type="radio"/>	<input type="radio"/>
If you embed the interests data into the users data instead of creating a separate document for each interest, you will increase the read operation costs for displaying the users and their associated interests	<input type="radio"/>	<input checked="" type="radio"/>

Box 1: Yes

Non-relational data increases write costs, but can decrease read costs.

Box 2: Yes

Non-relational data increases write costs, but can decrease read costs.

Box 3: No



Non-relational data increases write costs, but can decrease read costs.

---

## QUESTION 2

You have an Azure Cosmos DB for NoSQL account named account1.

You need to create a container named Container1 in account1 by using the Azure Cosmos DB .NET SDK. The solution must ensure that the items in Container1 never expire.

What should you set?

- A. TimeToLivePropertyPath to null
- B. TimeToLivePropertyPath to 0
- C. DefaultTimeToLive to null
- D. DefaultTimeToLive to -1

Correct Answer: D

Explanation:

Time to live for containers and items

The time to live value is set in seconds, and it is interpreted as a delta from the time that an item was last modified. You can set time to live on a container or an item within the container:

Time to Live on a container (set using DefaultTimeToLive):

If missing (or set to null), items are not expired automatically.

If present and the value is set to "-1", it is equal to infinity, and items don't expire by default.

If present and the value is set to some non-zero number "n" – items will expire "n" seconds after their last modified time.

Reference:

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/time-to-live>

---

## QUESTION 3

You have an Azure Cosmos DB for NoSQL account configured for global distribution across four regions.

At connection time, the SQL SDK needs to identify the optimal endpoint for reading and writing.

Which two factors can influence the SDK? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. the consistency level in RequestOptions
- B. regional latency



- C. the default consistency level
- D. the PreferredLocations configuration
- E. a region being available

Correct Answer: DE

Explanation:

Connecting to a preferred region using the API for NoSQL

The SDKs accept an optional parameter PreferredLocations that is an ordered list of Azure regions.

The SDK will automatically send all writes to the current write region. All reads will be sent to the first available region in the preferred locations list. If the request fails, the client will fail down the list to the next region.

The SDK will only attempt to read from the regions specified in preferred locations. So, for example, if the Azure Cosmos DB account is available in four regions, but the client only specifies two read(non-write) regions within the

PreferredLocations, then no reads will be served out of the read region that is not specified in PreferredLocations. If the read regions specified in the PreferredLocations list are not available, reads will be served out of write region.

Reference: <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/tutorial-global-distribution>

---

#### QUESTION 4

You have an Azure Cosmos DB Core (SQL) API account that is used by 10 web apps.

You need to analyze the data stored in the account by using Apache Spark to create machine learning models. The solution must NOT affect the performance of the web apps.

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

NOTE: Each correct selection is worth one point.

- A. In an Apache Spark pool in Azure Synapse, create a table that uses cosmos.olapas the data source.
- B. Create a private endpoint connection to the account.
- C. In an Azure Synapse Analytics serverless SQL pool, create a view that uses OPENROWSET and the CosmosDB provider.
- D. Enable Azure Synapse Link for the account and Analytical store on the container.
- E. In an Apache Spark pool in Azure Synapse, create a table that uses cosmos.oltpas the data source.

Correct Answer: AD

Explore analytical store with Apache Spark

1.

Navigate to the Data hub.



2.

Select the Linked tab (1), expand the Azure Cosmos DB group (if you don't see this, select the Refresh button above), then expand the WoodgroveCosmosDb account (2). Right-click on the transactions container (3), select New notebook (4), then select Load to DataFrame (5).

3.

In the generated code within Cell 1 (3), notice that the spark.read format is set to cosmos.olap. This instructs Synapse Link to use the container's analytical store. If we wanted to connect to the transactional store, like to read from the change feed or write to the container, we'd use cosmos.oltp instead.

Reference: <https://github.com/microsoft/MCW-Cosmos-DB-Real-Time-Advanced-Analytics/blob/main/Hands-on%20lab/HOL%20step-by%20step%20-%20Cosmos%20DB%20real-time%20advanced%20analytics.md>

## QUESTION 5

### HOTSPOT

You have an Azure Cosmos DB Core (SQL) API account named account1.

You have the Azure virtual networks and subnets shown in the following table.

Subnet	Network	IP address range	Virtual machine
subnet1	vnet1	10.0.0.0/24	VM1
subnet2	vnet1	10.0.1.0/24	VM2
subnet3	vnet2	10.1.0.0/24	VM3

The vnet1 and vnet2 networks are connected by using a virtual network peer.

The Firewall and virtual network settings for account1 are configured as shown in the exhibit.



Allow access from

All networks  Selected networks

Configure network security for your Azure Cosmos DB account. [Learn more.](#)

Virtual networks

Secure your Azure Cosmos DB account with virtual networks. [+ Add existing virtual network](#) [+Add new virtual network](#)

Virtual Network	Subnet	Address range	Endpoint Status
<input type="checkbox"/> vnet1 <input checked="" type="checkbox"/> vnet1	1 vnet1.subnet1	10.0.0.0/16 10.0.1.0/24	<input checked="" type="checkbox"/> Enabled

Firewall

Add IP ranges to allow access from the internet or your on-premises networks. [+Add my current IP](#) ⓘ

IP(Single IPv4 or CIDR range)

Exceptions

- Accept connections from within public Azure datacenters ⓘ
- Allow access from Azure Portal ⓘ

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
VM1 can access account 1	<input type="radio"/>	<input type="radio"/>
VM2 can access account 1	<input type="radio"/>	<input type="radio"/>
VM3 can access account 1	<input type="radio"/>	<input type="radio"/>

Correct Answer:



## Answer Area

Statements	Yes	No
VM1 can access account 1	<input checked="" type="radio"/>	<input type="radio"/>
VM2 can access account 1	<input type="radio"/>	<input checked="" type="radio"/>
VM3 can access account 1	<input type="radio"/>	<input checked="" type="radio"/>

Box 1: Yes

VM1 is on vnet1.subnet1 which has the Endpoint Status enabled.

Box 2: No

Only virtual network and their subnets added to Azure Cosmos account have access. Their peered VNets cannot access the account until the subnets within peered virtual networks are added to the account.

Box 3: No

Only virtual network and their subnets added to Azure Cosmos account have access.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-configure-vnet-service-endpoint>

### QUESTION 6

The following is a sample of a document in orders.





```
{
  "orderId" : "d4a91979b-5ead-43a3-b851-add9a71ac4b6",
  "customerId" : "f6e39103-bdc7-4346-9cfb-45daa4b2becf",
  "orderDate" : "2021-09-29",
  "orderItems" : [
    {
      "itemId" : "6c30412f-3cd7-4cab-813c-05942345720d",
      "name" : "blue pen",
      "type" : "pens",
      "count" : 10,
    },
    ...
  ],
  "total" : 12345,
  "status" : "ordered"
}
```

The orders container uses customerId as the partition key.

You need to provide a report of the total items ordered per month by item type. The solution must meet the following requirements:

1.

Ensure that the report can run as quickly as possible.

2.

Minimize the consumption of request units (RUs). What should you do?

- A. Configure the report to query orders by using a SQL query.
- B. Configure the report to query a new aggregate container. Populate the aggregates by using the change feed.
- C. Configure the report to query orders by using a SQL query through a dedicated gateway.
- D. Configure the report to query a new aggregate container. Populate the aggregates by using SQL queries that run daily.

Correct Answer: B

You can facilitate aggregate data by using Change Feed and Azure Functions, and then use it for reporting.

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

---

## QUESTION 7

You have a database named db1 in an Azure Cosmos DB for NoSQL account named account 1.



You need to write JSON data to db1 by using Azure Stream Analytics. The solution must minimize costs.

Which should you do before you can use db1 as an output of Stream Analytics?

- A. In account, add a private endpoint.
- B. In db1, create containers that have a custom indexing policy and analytical store disabled.
- C. In account, enable a dedicated gateway.
- D. In db1, create containers that have an automatic indexing policy and analytical store enabled.

Correct Answer: A

---

### QUESTION 8

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 9**

**HOTSPOT**

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

You need to use the Azure Cosmos DB SDK to replace a document by using optimistic concurrency.

What should you include in the code? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

## Answer Area

RequestOptions property to set:

	▼
AccessCondition	
ConsistencyLevel	
SessionToken	

Document property that will be compared:

	▼
_etag	
_id	
_rid	

Correct Answer:



## Answer Area

RequestOptions property to set:

	▼
AccessCondition	
ConsistencyLevel	
SessionToken	

Document property that will be compared:

	▼
_etag	
_id	
_rid	

Box 1: ConsistencyLevel

The ItemRequestOptions Class ConsistencyLevel property gets or sets the consistency level required for the request in the Azure Cosmos DB service.

Azure Cosmos DB offers 5 different consistency levels. Strong, Bounded Staleness, Session, Consistent Prefix and Eventual - in order of strongest to weakest consistency.

Box 2: \_etag

The ItemRequestOptions class helped us implement optimistic concurrency by specifying that we wanted the SDK to use the If-Match header to allow the server to decide whether a resource should be updated. The If-Match value is the ETag

value to be checked against. If the ETag value matches the server ETag value, the resource is updated.

Reference:

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.cosmos.itemrequestoptions>

<https://cosmosdb.github.io/labs/dotnet/labs/10-concurrency-control.html>

### QUESTION 10

#### HOTSPOT

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

You plan to create a container that will store employee data for 5,000 small businesses. Each business will have up to 25 employees. Each employee item will have an emailAddressvalue.



You need to ensure that the emailAddressvalue for each employee within the same company is unique.

To what should you set the partition key and the unique key? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

### Answer Area

Partition key

	▼
companyId	
companyId+emailAddress	
emailAddress	
employeeId	

Unique key

	▼
companyId	
emailAddress	
employeeId	

Correct Answer:

### Answer Area

Partition key

	▼
companyId	
companyId+emailAddress	
emailAddress	
employeeId	

Unique key

	▼
companyId	
emailAddress	
employeeId	



Box 1: CompanyID After you create a container with a unique key policy, the creation of a new or an update of an existing item resulting in a duplicate within a logical partition is prevented, as specified by the unique key constraint. The partition key combined with the unique key guarantees the uniqueness of an item within the scope of the container.

For example, consider an Azure Cosmos container with Email address as the unique key constraint and CompanyID as the partition key. When you configure the user's email address with a unique key, each item has a unique email address within a given CompanyID. Two items can't be created with duplicate email addresses and with the same partition key value.

Box 2: emailAddress

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

---

## QUESTION 11

### 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 12

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 13**

You have an Azure Cosmos DB for NoSQL account.

The change feed is enabled on a container named invoice.

You create an Azure function that has a trigger on the change feed.

What is received by the Azure function?

- A. all the properties of the updated items
- B. only the partition key and the changed properties of the updated items
- C. all the properties of the original items and the updated items
- D. only the changed properties and the system-defined properties of the updated items

Correct Answer: A

According to the Azure Cosmos DB documentation<sup>1</sup>, the change feed is a persistent record of changes to a container in the order they occur. The change feed outputs the sorted list of documents that were changed in the order in which they were modified.

The Azure function that has a trigger on the change feed receives all the properties of the updated items<sup>2</sup>. The change feed does not include the original items or only the changed properties. The change feed also includes some system-defined properties such as `_ts` (the last modified timestamp) and `_lsn` (the logical sequence number)<sup>3</sup>.

Therefore, the correct answer is "all the properties of the updated items"

[DP-420 PDF Dumps](#)

[DP-420 VCE Dumps](#)

[DP-420 Exam Questions](#)