

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.pass2lead.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 configure an Apache Kafka instance to ingest data from an Azure Cosmos DB Core (SQL) API account. The data from a container named telemetry must be added to a Kafka topic named `iot`. The solution must store the data in a

compact binary format.

Which three configuration items should you include in the solution? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. `"connector.class": "com.azure.cosmos.kafka.connect.source.CosmosDBSourceConnector"`
- B. `"key.converter": "org.apache.kafka.connect.json.JsonConverter"`
- C. `"key.converter": "io.confluent.connect.avro.AvroConverter"`
- D. `"connect.cosmos.containers.topicmap": "iot#telemetry"`
- E. `"connect.cosmos.containers.topicmap": "iot"`
- F. `"connector.class": "com.azure.cosmos.kafka.connect.source.CosmosDBSinkConnector"`

Correct Answer: CDF

C: Avro is binary format, while JSON is text.

F: Kafka Connect for Azure Cosmos DB is a connector to read from and write data to Azure Cosmos DB. The Azure Cosmos DB sink connector allows you to export data from Apache Kafka topics to an Azure Cosmos DB database. The connector polls data from Kafka to write to containers in the database based on the topics subscription.

D: Create the Azure Cosmos DB sink connector in Kafka Connect. The following JSON body defines config for the sink connector.

Extract:

```
"connector.class": "com.azure.cosmos.kafka.connect.sink.CosmosDBSinkConnector",  
  
"key.converter": "org.apache.kafka.connect.json.AvroConverter"  
  
"connect.cosmos.containers.topicmap": "hotels#kafka"
```

Incorrect Answers:

B: JSON is plain text.

Note, full example:

```
{ "name": "cosmosdb-sink-connector", "config": {  
  
  "connector.class": "com.azure.cosmos.kafka.connect.sink.CosmosDBSinkConnector",  
  
  "tasks.max": "1",
```

```
"topics": [  
  "hotels"  
],  
"value.converter": "org.apache.kafka.connect.json.AvroConverter",  
"value.converter.schemas.enable": "false",  
"key.converter": "org.apache.kafka.connect.json.AvroConverter",  
"key.converter.schemas.enable": "false",  
"connect.cosmos.connection.endpoint": "https://documents.azure.com:443/",  
"connect.cosmos.master.key": "",  
"connect.cosmos.databasename": "kafkaconnect",  
"connect.cosmos.containers.topicmap": "hotels#kafka"  
}]
```

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/kafka-connector-sink>

<https://www.confluent.io/blog/kafka-connect-deep-dive-converters-serialization-explained/>

QUESTION 2

You need to select the partition key for con-iot1. The solution must meet the IoT telemetry requirements. What should you select?

- A. the timestamp
- B. the humidity
- C. the temperature
- D. the device ID

Correct Answer: D

The partition key is what will determine how data is routed in the various partitions by Cosmos DB and needs to make sense in the context of your specific scenario. The IoT Device ID is generally the "natural" partition key for IoT applications.

Scenario: The iotdb database will contain two containers named con-iot1 and con-iot2. Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.

Reference: <https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/iot-using-cosmos-db>

QUESTION 3

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have an Azure Cosmos DB Core (SQL) API account named account 1 that uses autoscale throughput.

You need to run an Azure function when the normalized request units per second for a container in account1 exceeds a specific value.

Solution: You configure an Azure Monitor alert to trigger the function.

Does this meet the goal?

- A. Yes
- B. No

Correct Answer: A

You can set up alerts from the Azure Cosmos DB pane or the Azure Monitor service in the Azure portal.

Note: Alerts are used to set up recurring tests to monitor the availability and responsiveness of your Azure Cosmos DB resources. Alerts can send you a notification in the form of an email, or execute an Azure Function when one of your metrics reaches the threshold or if a specific event is logged in the activity log.

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

QUESTION 4

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: AB

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 5

You have an application that queries an Azure Cosmos DB for NoSQL account.

You discover that the following two queries run frequently,

```
SELECT * FROM c WHERE c.name = @name ORDER BY c.name  
DESC, c.timestamp DESC
```

```
SELECT * FROM c WHERE c.name = @name AND c.timestamp  
ORDER BY c.name ASC, c.timestamp ASC
```

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

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

Correct Answer: D

QUESTION 6

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 ConflictResolutionMode to Custom and you use the default settings for the policy.

Does this meet the goal?

- A. Yes
- B. No

Correct Answer: B

Setting ConflictResolutionMode to Custom and using the default settings for the policy will not ensure that conflicts are sent to the conflict feed. You need to define a custom stored procedure using the "conflictingItems" parameter to handle conflicts properly.

QUESTION 7

You are designing an Azure Cosmos DB Core (SQL) API solution to store data from IoT devices. Writes from the devices will occur every second. The following is a sample of the data.

```
{
  "id" : "03c1ca5a-db18-4231-908f-09a9bc7a7c3e",
  "deviceManufacturer" : "Contoso, Ltd",
  "deviceId" : "f460df85-799f-4d58-b051-67561b4993c6",
  "timestamp" : "2021-09-19T13:47:45",
  "sensor1Value" : true,
  "sensor2Value" : "75",
  "sensor3Value" : "4554",
  "sensor4Value" : "454",
  "sensor5Value" : "42128"
}
```

You need to select a partition key that meets the following requirements for writes:

1.

Minimizes the partition skew

2.

Avoids capacity limits

3.

Avoids hot partitions What should you do?

A. Use timestamp as the partition key.

B. Create a new synthetic key that contains deviceId and sensor1Value.

C. Create a new synthetic key that contains deviceId and deviceManufacturer.

D. Create a new synthetic key that contains deviceId and a random number.

Correct Answer: D

Use a partition key with a random suffix. Distribute the workload more evenly is to append a random number at the end of the partition key value. When you distribute items in this way, you can perform parallel write operations across partitions.

Incorrect Answers:

A: You will also not like to partition the data on "DateTime", because this will create a hot partition. Imagine you have partitioned the data on time, then for a given minute, all the calls will hit one partition. If you need to retrieve the data for a customer, then it will be a fan-out query because data may be distributed on all the partitions.

B: Sensor1Value has only two values.

C: All the devices could have the same manufacturer.

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

QUESTION 8

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 9

You are troubleshooting the current issues caused by the application updates.

Which action can address the application updates issue without affecting the functionality of the application?

- A. Enable time to live for the con-product container.
- B. Set the default consistency level of account1 to strong.
- C. Set the default consistency level of account1 to bounded staleness.

D. Add a custom indexing policy to the con-product container.

Correct Answer: C

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.

Scenario: Application updates in con-product frequently cause HTTP status code 429 "Too many requests". You discover that the 429 status code relates to excessive request unit (RU) consumption during the updates.

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

QUESTION 10

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

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

You need to make the contents of container1 available as reference data for an Azure Stream Analytics job.

Solution: You create an Azure Data Factory pipeline that uses Azure Cosmos DB Core (SQL) API as the input and Azure Blob Storage as the output.

Does this meet the goal?

A. Yes

B. No

Correct Answer: B

Instead create an Azure function that uses Azure Cosmos DB Core (SQL) API change feed as a trigger and Azure event hub as the output.

The Azure Cosmos DB change feed is a mechanism to get a continuous and incremental feed of records from an Azure Cosmos container as those records are being created or modified. Change feed support works by listening to container for any changes. It then outputs the sorted list of documents that were changed in the order in which they were modified.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/changefeed-ecommerce-solution>

QUESTION 11

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 documentation¹, 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 items². 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)³.

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

QUESTION 12

HOTSPOT

You have three containers in an Azure Cosmos DB Core (SQL) API account as shown in the following table.

Name	Database	Time to Live
cn1	db1	On (no default)
cn2	db1	Off
cn3	db1	On (no default)

You have the following Azure functions:

1.

A function named Fn1 that reads the change feed of cn1

2.

A function named Fn2 that reads the change feed of cn2

3.

A function named Fn3 that reads the change feed of cn3 You perform the following actions:

1.

Delete an item named item1 from cn1.

2.

Update an item named item2 in cn2.

3.

For an item named item3 in cn3, update the item time to live to 3,600 seconds.

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
Fn1 will receive item1 from the change feed	<input type="radio"/>	<input type="radio"/>
Fn2 can check the <code>_etag</code> of item2 to see whether the item is an update or an insert	<input type="radio"/>	<input type="radio"/>
Fn3 will receive item3 from the change feed	<input type="radio"/>	<input type="radio"/>

Correct Answer:

Answer Area

Statements	Yes	No
Fn1 will receive item1 from the change feed	<input type="radio"/>	<input checked="" type="radio"/>
Fn2 can check the <code>_etag</code> of item2 to see whether the item is an update or an insert	<input type="radio"/>	<input checked="" type="radio"/>
Fn3 will receive item3 from the change feed	<input checked="" type="radio"/>	<input type="radio"/>

Box 1: No

Azure Cosmos DB's change feed is a great choice as a central data store in event sourcing architectures where all data ingestion is modeled as writes (no updates or deletes).

Note: The change feed does not capture deletes. If you delete an item from your container, it is also removed from the change feed. The most common method of handling this is adding a soft marker on the items that are being deleted. You

can add a property called "deleted" and set it to "true" at the time of deletion. This document update will show up in the change feed. You can set a TTL on this item so that it can be automatically deleted later.

Box 2: No

The `_etag` format is internal and you should not take dependency on it, because it can change anytime.

Box 3: Yes

Change feed support in Azure Cosmos DB works by listening to an Azure Cosmos container for any changes.

Reference:

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

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

QUESTION 13

You plan to create an Azure Cosmos DB Core (SQL) API account that will use customer-managed keys stored in Azure Key Vault.

You need to configure an access policy in Key Vault to allow Azure Cosmos DB access to the keys.

Which three permissions should you enable in the access policy? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Wrap Key
- B. Get
- C. List
- D. Update
- E. Sign
- F. Verify
- G. Unwrap Key

Correct Answer: ABG

To Configure customer-managed keys for your Azure Cosmos account with Azure Key Vault:

Add an access policy to your Azure Key Vault instance:

1.

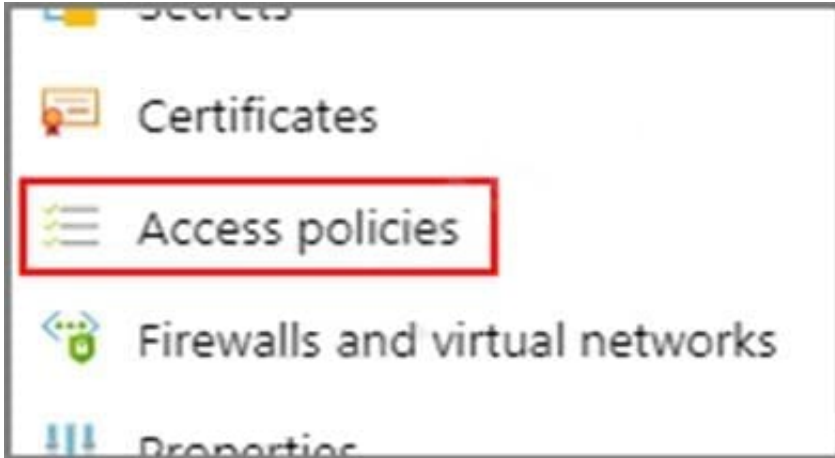
From the Azure portal, go to the Azure Key Vault instance that you plan to use to host your encryption keys. Select Access Policies from the left menu:

2.

Select + Add Access Policy.

3.

Under the Key permissions drop-down menu, select Get, Unwrap Key, and Wrap Key permissions:



Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-setup-cmk>

QUESTION 14

You have the following query.

```
SELECT * FROM ? WHERE c.sensor = "TEMP1" AND c.value = 1619146031231
```

You need to recommend a composite index strategy that will minimize the request units (RUs) consumed by the query.

What should you recommend?

- A. a composite index for (sensor ASC, value ASC) and a composite index for (sensor ASC, timestamp ASC)
- B. a composite index for (sensor ASC, value ASC, timestamp ASC) and a composite index for (sensor DESC, value DESC, timestamp DESC)
- C. a composite index for (value ASC, sensor ASC) and a composite index for (timestamp ASC, sensor ASC)

D. a composite index for (sensor ASC, value ASC, timestamp ASC)

Correct Answer: A

If a query has a filter with two or more properties, adding a composite index will improve performance.

Consider the following query:

```
SELECT * FROM c WHERE c.name = "Tim" and c.age > 18
```

In the absence of a composite index on (name ASC, and age ASC), we will utilize a range index for this query. We can improve the efficiency of this query by creating a composite index for name and age.

Queries with multiple equality filters and a maximum of one range filter (such as >,