



**Vendor:** Microsoft

**Exam Code:** DP-420

**Exam Name:** Designing and Implementing Cloud-Native Applications Using Microsoft Azure Cosmos DB

**Version:** DEMO

## QUESTION 1

### Case Study 1 - Litware, inc

#### Overview

Litware, Inc. is a United States-based grocery retailer. Litware has a main office and a primary datacenter in Seattle. The company has 50 retail stores across the United States and an emerging online presence. Each store connects directly to the internet. Existing environment. Cloud and Data Service Environments. Litware has an Azure subscription that contains the resources shown in the following table.

Name	Type	Description
account1	Azure Cosmos DB Core (SQL) API account	The account has a single read-write region and contains a database named productdb. The productdb database contains two containers named con-product and con-productVendor. The account uses the session default consistency level.
iothub1	Azure IoT hub	The IoT hub collects per-second temperature and humidity telemetry from 5,000 IoT devices in the retail stores. A single telemetry reading generates 1 KB of data.
streamanalytics1	Azure Stream Analytics job	The job processes telemetry data collected from iothub1.

Each container in productdb is configured for manual throughput. The con-product container stores the company's product catalog data. Each document in con-product includes a con-productVendor value. Most queries targeting the data in con-product are in the following format.  
 SELECT \* FROM con-product p WHERE p.con-productVendor = 'name'  
 Most queries targeting the data in the con-productVendor container are in the following format  
 SELECT \* FROM con-productVendor pv  
 ORDER BY pv.creditRating, pv.yearFounded

#### Existing environment.

##### Current Problems.

Litware identifies the following issues:

- Updates to product categories in the con-productVendor container do not propagate automatically to documents in the con-product container.
- 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.

#### Requirements.

##### Planned Changes

Litware plans to implement a new Azure Cosmos DB Core (SQL) API account named account2 that will contain a database named iotdb. The iotdb database will contain two containers named con-iot1 and con-iot2.

Litware plans to make the following changes:

- Store the telemetry data in account2.
- Configure account1 to support multiple read-write regions.
- Implement referential integrity for the con-product container.
- Use Azure Functions to send notifications about product updates to different recipients.
- Develop an app named App1 that will run from all locations and query the data in account1.
- Develop an app named App2 that will run from the retail stores and query the data in account2.

- App2 must be limited to a single DNS endpoint when accessing account2.

**Requirements.**

**Business Requirements**

Litware identifies the following business requirements:

- 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.
- Ensure that Azure Cosmos DB costs for IoT-related processing are predictable.
- Minimize the number of firewall changes in the retail stores.

**Requirements.**

**Product Catalog Requirements**

Litware identifies the following requirements for the product catalog:

- Implement a custom conflict resolution policy for the product catalog data.
- Minimize the frequency of errors during updates of the con-product container.
- Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.
- Trigger the execution of two Azure functions following every update to any document in the con-product container.

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.

**Answer: D**

**Explanation:**

For 429's on updating documents it is recommended to create a custom index policy, and only contain the properties that are needed.

<https://learn.microsoft.com/en-us/azure/cosmos-db/sql/troubleshoot-request-rate-too-large?tabs=resource-specific#429s-on-create-replace-or-upsert-document-requests>

**QUESTION 2**

You have an application named App1 that reads the data in an Azure Cosmos DB Core (SQL) API account. App1 runs the same read queries every minute. The default consistency level for the account is set to eventual.

You discover that every query consumes request units (RUs) instead of using the cache.

You verify the IntegratedCacheItemHitRate metric and the IntegratedCacheQueryHitRate metric.

Both metrics have values of 0.

You verify that the dedicated gateway cluster is provisioned and used in the connection string.

You need to ensure that App1 uses the Azure Cosmos DB integrated cache.

What should you configure?

- A. the indexing policy of the Azure Cosmos DB container
- B. the consistency level of the requests from App1
- C. the connectivity mode of the App1 CosmosClient
- D. the default consistency level of the Azure Cosmos DB account

**Answer: C**

**Explanation:**

Because the integrated cache is specific to your Azure Cosmos DB account and requires significant CPU and memory, it requires a dedicated gateway node. Connect to Azure Cosmos DB using gateway mode.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/integrated-cache-faq>

**QUESTION 3**

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
```

**Answer: BD**

**Explanation:**

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 4**

You are designing an Azure Cosmos DB Core (SQL) API solution to store data from IoT devices. Writes from the devices will be 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:

- Minimizes the partition skew
- Avoids capacity limits
- 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.

**Answer: D**

**Explanation:**

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 5

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

You run the following query against a container in the account.

```
SELECT
IS_NUMBER("1234") AS A,
IS_NUMBER(1234) AS B,
IS_NUMBER({prop: 1234}) AS C
```

What is the output of the query?

- A. [{"A": false, "B": true, "C": false}]
- B. [{"A": true, "B": false, "C": true}]
- C. [{"A": true, "B": true, "C": false}]
- D. [{"A": true, "B": true, "C": true}]

**Answer: A**

**Explanation:**

IS\_NUMBER returns a Boolean value indicating if the type of the specified expression is a number.

"1234" is a string, not a number.

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

### QUESTION 7

You need to implement a trigger in Azure Cosmos DB Core (SQL) API that will run before an item is inserted into a container.

Which two actions should you perform to ensure that the trigger runs? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Append pre to the name of the JavaScript function trigger.
- B. For each create request, set the access condition in RequestOptions.
- C. Register the trigger as a pre-trigger.
- D. For each create request, set the consistency level to session in RequestOptions.

E. For each create request, set the trigger name in RequestOptions.

**Answer:** CE

**Explanation:**

C: When triggers are registered, you can specify the operations that it can run with.

E: When executing, pre-triggers are passed in the RequestOptions object by specifying PreTriggerInclude and then passing the name of the trigger in a List object.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-use-stored-procedures-triggers-udfs>

### QUESTION 8

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

You need to create an Azure Monitor query that lists recent modifications to the regional failover policy.

Which Azure Monitor table should you query?

- A. CDBControlPlaneRequests
- B. CDBQueryRunTimeStatistics
- C. CDBPartitionKeyStatistics
- D. CDBDataPlaneRequests

**Answer:** A

**Explanation:**

The CDBControlPlaneRequests table details all control plane operations executed on the account, which include modifications to the regional failover policy, indexing policy, IAM role assignments, backup/restore policies, VNet and firewall rules, private links as well as updates and deletes of the account.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-monitor/reference/tables/cdbcontrolplanerequests>

### QUESTION 9

You need to create a database in an Azure Cosmos DB Core (SQL) API account. The database will contain three containers named coll1, coll2, and coll3. The coll1 container will have unpredictable read and write volumes. The coll2 and coll3 containers will have predictable read and write volumes. The expected maximum throughput for coll1 and coll2 is 50,000 request units per second (RU/s) each.

How should you provision the collection while minimizing costs?

- A. Create a serverless account.
- B. Create a provisioned throughput account. Set the throughput for coll1 to Autoscale. Set the throughput for coll2 and coll3 to Manual.
- C. Create a provisioned throughput account. Set the throughput for coll1 to Manual. Set the throughput for coll2 and coll3 to Autoscale.

**Answer:** B

**Explanation:**

Manual is best suited for workloads with steady or predictable traffic.

Autoscale workloads is best suited with variable or unpredictable traffic.

Note: Azure Cosmos DB allows you to set provisioned throughput on your databases and containers. There are two types of provisioned throughput, standard (manual) or autoscale.

Autoscale provisioned throughput in Azure Cosmos DB allows you to scale the throughput (RU/s) of your database or container automatically and instantly. The throughput is scaled based on the



usage, without impacting the availability, latency, throughput, or performance of the workload.

The use cases of autoscale include:

\* Variable or unpredictable workloads

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-choose-offer>

## QUESTION 10

### Hotspot Question

You configure a backup for an Azure Cosmos DB Core (SQL) API account as shown in the following exhibit.

**Settings**

- Default consistency
- Backup & Restore**
- Firewall and virtual networks
- Private Endpoint Connections
- CORS
- Keys
- Advisor recommendations
- Add Azure Cognitive Search
- Add Azure Function
- Advanced security (preview)

**Backup Interval**  
How often would you like your backups to be performed?  
120 Minute(s)  
60-1440

**Backup Retention**  
How long would you like your backups to be saved?  
12 Hours(s)  
8-720

**Copies of data retained**  
By default, Azure Cosmos DB backups 2 copies of data for free. charged based on the [pricing details here](#)

**Backup storage redundancy**  
☒ Geo-redundant backup storage  
☐ Zone-redundant backup storage  
☐ Locally-redundant backup storage

Use the drop-down menus to select the answer choice that completes each statement based on the information presented in the graphic.

NOTE: Each correct selection is worth one point.

### Answer Area

The current backup interval and retention policy provide protection for **[answer choice]**

	▼
1 hour	
2 hours	
6 hours	
12 hours	
3 days	

In case of emergency, you must **[answer choice]** to restore the backup

	▼
create a support ticket	
use the Backup & Restore settings	
use the Point in Time Restore settings	



**Answer:**

**Answer Area**

The current backup interval and retention policy provide protection for [answer choice]

	▼
1 hour	
2 hours	
6 hours	
12 hours	
3 days	

In case of emergency, you must [answer choice] to restore the backup

	▼
create a support ticket	
use the Backup & Restore settings	
use the Point in Time Restore settings	

**Explanation:**

Box 1: 12 hours

The number of copies stored depends on the interval and the retention period.

The default retention backup interval is 240 minutes (4h) and the retention period is 8h, that's why 2 copies are stored by default. The first 2 copies are free as well.

In this example, every 2h a copy is created and is kept for 12h. This means there can be up to 6 copies. 4 of the 6 copies are billed according to Azure blob storage pricing.

Box 2: create a support ticket

Request data restore from a backup

If you accidentally delete your database or a container, you can file a support ticket or call the Azure support to restore the data from automatic online backups.

Note: With Azure Cosmos DB SQL API accounts, you can also maintain your own backups by using one of the following approaches:

Use Azure Data Factory to move data periodically to a storage of your choice.

Use Azure Cosmos DB change feed to read data periodically for full backups or for incremental changes, and store it in your own storage.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/configure-periodic-backup-restore>

**QUESTION 11**

You have an Azure Cosmos DB Core (SQL) API account that has multiple write regions.

You need to receive an alert when requests that target the database exceed the available request units per second (RU/s).

Which Azure Monitor signal should you use?

- A. Data Usage
- B. Metadata Requests
- C. Region Removed
- D. Region Added

**Answer: B**

**Explanation:**

Metric Metadata Requests: Count of metadata requests. Azure Cosmos DB maintains system

metadata container for each account, that allows you to enumerate collections, databases, etc., and their configurations, free of charge.

Used to monitor throttles due to metadata requests.

Incorrect:

Not A: Metric Data Usage: Total data usage reported at 5-minutes granularity per region. Used to monitor total data usage at container and region, minimum granularity should be 5 minutes.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/monitor-cosmos-db-reference#metrics>

## QUESTION 12

**Note:** This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

**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 database in an Azure Cosmos DB Core (SQL) API 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 conflicts are sent to the conflicts 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

**Answer:** A

**Explanation:**

There are three ways for conflict resolution with the SDK:

1. Last-Writer-Wins: Conflicts do NOT show up in conflict feed
2. Custom Conflict Resolution Policy with Stored Proc: Conflicts do NOT show up in conflict feed unless there's an error in your stored procedure
3. Custom Conflict Resolution Policy: Conflicts DO show up in conflict feed.

<https://learn.microsoft.com/en-us/azure/cosmos-db/sql/how-to-manage-conflicts?tabs=dotnetv2%2Capi-async%2Casync#create-a-custom-conflict-resolution-policy>

## Thank You for Trying Our Product

### Passleader Certification Exam Features:

- ★ More than **99,900** Satisfied Customers Worldwide.
- ★ Average **99.9%** Success Rate.
- ★ **Free Update** to match latest and real exam scenarios.
- ★ **Instant Download** Access! No Setup required.
- ★ Questions & Answers are downloadable in **PDF** format and **VCE** test engine format.
- ★ Multi-Platform capabilities - **Windows, Laptop, Mac, Android, iPhone, iPod, iPad**.
- ★ **100%** Guaranteed Success or **100%** Money Back Guarantee.
- ★ **Fast**, helpful support **24x7**.



View list of all certification exams: <http://www.passleader.com/all-products.html>



Microsoft



ORACLE



CITRIX



JUNIPER  
NETWORKS



EMC<sup>2</sup>  
where information lives

**10% Discount Coupon Code: ASTR14**