We offer free update service for one year!
https://www.certqueen.com

Exam : **DP-420**

Title : Designing and Implementing Cloud-Native Applications Using Microsoft Azure Cosmos DB

Version : DEMO

1. Topic 1: Litware, inc

Case Study
This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study
To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

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: C
Explanation:
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

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
Answer: D
Explanation:
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

3.You need to identify which connectivity mode to use when implementing App2. The solution must support the planned changes and meet the business requirements.
Which connectivity mode should you identify?
A. Direct mode over HTTPS
B. Gateway mode (using HTTPS)
C. Direct mode over TCP
Answer: C
Explanation:
Scenario: 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.
By using Azure Private Link, you can connect to an Azure Cosmos account via a private endpoint. The private endpoint is a set of private IP addresses in a subnet within your virtual network.
When you're using Private Link with an Azure Cosmos account through a direct mode connection, you can use only the TCP protocol. The HTTP protocol is not currently supported.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-configure-private-endpoints

4.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-product Vendor containers.
Answer: D
Explanation:
App1 queries the con-product and con-product Vendor 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.
Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/consistency-levels

5.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 leaseCollectionNair.e
C. Configure the trigger for each function to use a different leaseCollectionName
D. Configure the trigger for each function to use the same leaseCollectionPrefix
Answer: A, C
Explanation:
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