



- Expert Verified, Online, **Free**.



CERTIFICATION TEST

- CertificationTest.net - Cheap & Quality Resources With Best Support

HOTSPOT -

You have an Azure Cosmos DB Core (SQL) API account named account1 that has the disableKeyBasedMetadataWriteAccess property enabled. You are developing an app named App1 that will be used by a user named DevUser1 to create containers in account1. DevUser1 has a non-privileged user account in the Azure Active Directory (Azure AD) tenant.

You need to ensure that DevUser1 can use App1 to create containers in account1.

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

NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

Grant permissions to create containers by using:

	▼
Account keys	
Resource tokens	
Role-based access control (RBAC)	

Create containers by using the:

	▼
Azure AD Graph API	
Azure Resource Manager API	
SQL (Core) API	

Suggested Answer:**Answer Area**

Grant permissions to create containers by using:

	▼
Account keys	
Resource tokens	
Role-based access control (RBAC)	

Create containers by using the:

	▼
Azure AD Graph API	
Azure Resource Manager API	
SQL (Core) API	

Box 1: Resource tokens -

Resource tokens provide access to the application resources within a database. Resource tokens:

Provide access to specific containers, partition keys, documents, attachments, stored procedures, triggers, and UDFs.

Box 2: Azure Resource Manager API

You can use Azure Resource Manager to help deploy and manage your Azure Cosmos DB accounts, databases, and containers.

Incorrect Answers:

The Microsoft Graph API is a RESTful web API that enables you to access Microsoft Cloud service resources.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/secure-access-to-data> <https://docs.microsoft.com/en-us/rest/api/resources/>

 **lakime**  2 years, 8 months ago

I think it will be "Role-based access control" as Resource Token doesn't cooperate with AD, regarding second part - ARM is correct
upvoted 23 times

 **nkav**  2 years, 6 months ago

RBAC is the answer.

upvoted 6 times



  **practicewizards** Most Recent 2 weeks, 3 days ago

The correct answer is:

RBAC

ARM



upvoted 1 times

  **WimTS** 2 months, 2 weeks ago

RBAC to grant access to the user



The user will use App1 to create the container, so the app will use the SQL (Core) API

upvoted 1 times

  **anto69** 3 months, 2 weeks ago

RBAC and AzureRM API

upvoted 1 times

  **Tuopikson** 4 months, 3 weeks ago

Correct selections are:

Role-Based Access Control (RBAC)

SQL (Core) API

RBAC allows you to manage access to Azure resources, ensuring that DevUser1 has the necessary permissions to create containers in the Azure Cosmos DB Core (SQL) API account. Once permissions are granted, DevUser1 can use the SQL (Core) API to create containers.

upvoted 1 times



  **rakun** 7 months, 4 weeks ago

first is RBAC:

Azure Cosmos DB RBAC is the ideal access control method in situations where:

- You want to use Microsoft Entra identities to authenticate your requests.

upvoted 1 times

  **3a0b61c** 9 months, 2 weeks ago

RBAC/ARM

<https://learn.microsoft.com/en-us/azure/cosmos-db/audit-control-plane-logs#disable-key-based-metadata-write-access>

□After you set this property, changes to any resource can happen from a user with the proper Azure role and credentials.

□After the disableKeyBasedMetadataWriteAccess is turned on, if the SDK based clients run create or update operations, an error "Operation 'POST' on resource 'ContainerNameorDatabaseName' is not allowed through Azure Cosmos DB endpoint is returned. You have to turn on access to such operations for your account, or perform the create/update operations through Azure Resource Manager, Azure CLI or Azure PowerShell.

upvoted 2 times

  **[Removed]** 1 year, 2 months ago

RBAC

SQL API

upvoted 1 times

  **XiangRongChang** 1 year, 5 months ago

For me is Azure Resource Manager API.

When disableKeyBasedMetadataWriteAccess is set to true, the metadata operations issued by the SDK are blocked. Alternatively, you can use Azure portal, Azure CLI, Azure PowerShell, or Azure Resource Manager template deployments to perform these operations.

upvoted 1 times

  **xRiot007** 12 months ago

"When disableKeyBasedMetadataWriteAccess is set to true, the metadata operations issued by the SDK are blocked." - unless the user has an AD account (he does) with proper access rights (Cosmos DB Contributor) - this is not specified.

upvoted 1 times

  **azuredemo2022three** 1 year, 6 months ago

Correct Answer Should be.

Grant permission to create containers by using: Role-based access control (RBAC)

Create containers by using: SQL (Core) API

upvoted 5 times

  **essdeecee** 2 years, 3 months ago

More likely to be SQL (Core) API. Permission for Cosmos is required whereas Azure Resource Manager would need portal permissions.

upvoted 3 times

HOTSPOT -

You have an Azure Cosmos DB Core (SQL) account that has a single write region in West Europe.

You run the following Azure CLI script.

```
az cosmosdb update -n $accountName -g $resourceGroupName \
  --locations regionName='West Europe' failoverPriority=0 isZoneRedundant=False \
  --locations regionName='North Europe' failoverPriority=1 isZoneRedundant=False

az cosmosdb failover-priority-change -n $accountName -g $resourceGroupName \
  --failover-policies 'North Europe=0' 'West Europe=1'
```

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
After running the script, there will be an instance of Azure Cosmos DB in North Europe that is writable	<input type="radio"/>	<input type="radio"/>
After running the script, the Azure Cosmos DB instance in West Europe will be writable	<input type="radio"/>	<input type="radio"/>
The cost of the Azure Cosmos DB account is unaffected by running the script	<input type="radio"/>	<input type="radio"/>

Suggested Answer:

Answer Area

Statements	Yes	No
After running the script, there will be an instance of Azure Cosmos DB in North Europe that is writable	<input checked="" type="radio"/>	<input type="radio"/>
After running the script, the Azure Cosmos DB instance in West Europe will be writable	<input type="radio"/>	<input checked="" type="radio"/>
The cost of the Azure Cosmos DB account is unaffected by running the script	<input checked="" type="radio"/>	<input type="radio"/>

Box 1: Yes -

The Automatic failover option allows Azure Cosmos DB to failover to the region with the highest failover priority with no user action should a region become unavailable.

Box 2: No -

West Europe is used for failover. Only North Europe is writable.



To Configure multi-region set UseMultipleWriteLocations to true.

Box 3: Yes -

Provisioned throughput with single write region costs \$0.008/hour per 100 RU/s and provisioned throughput with multiple writable regions costs \$0.016/per hour per 100 RU/s.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-multi-master> <https://docs.microsoft.com/en-us/azure/cosmos-db/optimize-cost-regions>

 **ngthien041292**  2 years, 2 months ago

Vote YNN

upvoted 26 times

 **wispa2001**  2 years, 1 month ago

1:No

2:Yes

3:No

upvoted 12 times

  **ognamala** 1 year, 10 months ago

Your first answer is incorrect - in the above, there is a change of the priority zero, effectively changing the write region.

upvoted 1 times

  **ognamala** 1 year, 10 months ago

Your second answer is also incorrect - again the priority 0 is changing which initiates manual failover - so that should be No.

upvoted 1 times

  **Ranzzan** 1 year, 4 months ago

Should be YNY. for the third one tried it on pricing calculator, the cost does not change.

upvoted 2 times

  **KELCHOSPENSE** Most Recent 2 weeks, 1 day ago

- After running the script, there will be an instance of Azure Cosmos DB in North Europe that is writable. – Yes

- After running the script, the Azure Cosmos DB instance in West Europe will be writable. – No

- The cost of the Azure Cosmos DB account is unaffected by running the script. – No

upvoted 1 times

  **practicewizards** 2 weeks, 3 days ago

Y N Y

- When priority zero it is writtable.

- West Europe switched to Read Region.

- Since only one write region, the cost is the same.

upvoted 1 times

  **anto69** 3 months, 2 weeks ago

Y-N-N (ChatGPT verified with correct explanation)

upvoted 1 times

  **YellowSky002** 5 months, 2 weeks ago

If the account has a single write, how do these commands impact the number of databases or writable ones? There is ONLY one primary database , there is no other database.

upvoted 1 times

  **[Removed]** 8 months ago



Answer is YNN

upvoted 1 times

  **60919_S** 9 months, 3 weeks ago

Which is the right answer totally confused .

upvoted 2 times

  **Shiven** 10 months, 2 weeks ago

1-Y

2-N

3-Y

failover_priority - (Required) The failover priority of the region. A failover priority of 0 indicates a write region. The maximum value for a failover priority = (total number of regions - 1). Failover priority values must be unique for each of the regions in which the database account exists. Changing this causes the location to be re-provisioned and cannot be changed for the location with failover priority 0.

upvoted 3 times

  **orionduo** 11 months ago

The answer are correct.

A failover priority of 0 indicates a write region.

The price of West Euro and North Euro for Cosmos DB is just exactly the same. You may check it with Azure Pricing Calculator:

<https://azure.microsoft.com/en-us/pricing/calculator/>

upvoted 1 times

🗳️ 👤 **azuredemo2022three** 1 year ago

Answer should be

No - After running the script, there will not be an instance of Azure Cosmos DB in North Europe that is writable. The script sets the failover priority of North Europe to 0, indicating that it is a secondary region and not writable.

Yes - After running the script, the Azure Cosmos DB instance in West Europe will still be writable. The script does not change the failover priority or zone redundancy of the West Europe region, so it remains the primary region and writable.

Yes - The cost of the Azure Cosmos DB account is unaffected by running the script. The script only modifies the configuration and failover policies of the account, but it does not change the pricing or billing aspects of the account.

upvoted 1 times

🗳️ 👤 **azuredemo2022three** 1 year ago

Yes, Answer is YNN

upvoted 1 times

🗳️ 👤 **kel7718** 1 year, 6 months ago

A failover priority of 0 indicates a write region (<https://learn.microsoft.com/en-us/rest/api/cosmos-db-resource-provider/2021-04-01-preview/database-accounts/failover-priority-change?tabs=HTTP>)

Knowing this, North Europe is now 0 so write region, first answer is thus Yes. West Europe is now 1 and we did not enable multiple write regions so we can no longer write to west europe and costs haven't increased as it is still a single write region. YNN

upvoted 1 times

🗳️ 👤 **Alex22022** 1 year, 5 months ago

Costs have increased since North Europe is added. Now we have two regions instead of one. Each region is billed: <https://learn.microsoft.com/en-us/training/modules/configure-replication-manage-failovers-azure-cosmos-db/4-evaluate-cost-distributing-data-globally>

But since the statment is 'The cost ... is unaffected ...', the answer is no.

Therefore: YNN

upvoted 2 times

🗳️ 👤 **codingdown** 1 year, 7 months ago

bah , not clear at all

upvoted 2 times

🗳️ 👤 **ognamala** 1 year, 10 months ago

Yes - triggering a manual failover by changing priority zero causes write and read regions to be swapped

No - triggering a manual failover by changing priority zero causes write and read regions to be swapped

Yes - unaffected only because we are switching between west and north europe, if it was say switzerland it would be more expensive

upvoted 7 times

🗳️ 👤 **SQLK** 1 year, 10 months ago

Why option C is yes??

upvoted 1 times

🗳️ 👤 **Alex22022** 1 year, 5 months ago

C is No because each region is billed additionally. Equivalent emxample from the docs: <https://learn.microsoft.com/en-us/training/modules/configure-replication-manage-failovers-azure-cosmos-db/4-evaluate-cost-distributing-data-globally>

upvoted 1 times

🗳️ 👤 **susejzopol** 1 year, 7 months ago

I think because they are un the same región. This cause that cost doesn't increase.

upvoted 1 times

🗳️ 👤 **Gtus** 2 years ago

The third statement here is tricky. Normally, Azure resource price is different for regions, this applicable to Cosmos DB as well. However, the price of West Euro and North Euro for Costome DB is just exactly the same. You may check it with Azure Pricing Calculator: <https://azure.microsoft.com/en-us/pricing/calculator/>

upvoted 4 times

🗳️ 👤 **lakime** 2 years, 2 months ago

IMO C should be No

upvoted 3 times

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
```

Suggested 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>

 **itsmenida1** Highly Voted 2 years ago

correct

upvoted 6 times

 **practicewizards** Most Recent 2 weeks, 3 days ago

The goal is to show top five orders. Most orders sorted descendent.



B D

upvoted 1 times

  **[Removed]** 10 months ago

BD, these are the queries that have the correct ORDER BY

upvoted 1 times

  **dudud** 1 year, 3 months ago

I think that on top in the grid with data there should be Types and not just Type since in all queries use .types

upvoted 1 times

  **virgilpza** 1 year, 10 months ago

answer is correct

upvoted 3 times

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 emailAddress value.

You need to ensure that the emailAddress value 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	

Answer Area

Partition key

	▼
companyId	
companyId+emailAddress	
emailAddress	
employeeId	

Suggested Answer:

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>

 **ognamala** Highly Voted 1 year, 4 months ago

Solution is correct - companyId for partition key, and emailAddress for unique key

Additional justification why you would definitely set the partition key to companyId, is by also keeping potential queries in mind, typically in this scenario you will always end up with an equality filter on the companyId, hence it makes sense to partition by companyId, which will result in better query performance

UniqueKey will need to be set to emailAddress, since the job of a unique key is to ensure uniqueness of a value within a logical partition
<https://docs.microsoft.com/en-us/azure/cosmos-db/unique-keys>

upvoted 12 times

  **practicewizards** Most Recent 2 weeks, 3 days ago

it is correct:

- CompanyId for partition.
- EmailAddress for Unique key.

upvoted 1 times



  **azuredemo2022three** 6 months, 4 weeks ago

The given answer is correct

Partition Key: CompanyId

Unique Key: emailAddress

upvoted 2 times

  **lakime** 1 year, 8 months ago

Correct - Logical partitions are formed based on the value of a partition key that is associated with each item in a container. All the items in a logical partition have the same partition key value.

So segregating by companyid makes sense

E-mail will be unique

upvoted 2 times

HOTSPOT -

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account. The container1 container has 120 GB of data. The following is a sample of a document in container1.

```
{
  "customerId" : "5425",
  "orderId" : "9d7816e6-f401-42ba-ad05-0e03de35c0b8",
  "orderDate" : "2019-05-03",
  "orderDetails" : []
}
```

The orderId property is used as the partition key.

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 run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.orderDate = "2019-05-03"</pre>	<input type="radio"/>	<input type="radio"/>
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.customerId = "5425"</pre>	<input type="radio"/>	<input type="radio"/>
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.orderDate = "2019-05-03" and c.orderId = "9d7816e6-f401-42ba-ad05-0e03de35c0b8"</pre>	<input type="radio"/>	<input type="radio"/>

Suggested Answer:

Answer Area

Statements	Yes	No
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.orderDate = "2019-05-03"</pre>	<input checked="" type="radio"/>	<input type="radio"/>
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.customerId = "5425"</pre>	<input checked="" type="radio"/>	<input type="radio"/>
If you run the following query, the query will run as a cross-partition query <pre>SELECT * FROM c where c.orderDate = "2019-05-03" and c.orderId = "9d7816e6-f401-42ba-ad05-0e03de35c0b8"</pre>	<input type="radio"/>	<input checked="" type="radio"/>

Box 1: Yes -


Records with different OrderIDs will match.

Box 2: Yes -

Records with different OrderIDs will match.

Box 3: No -

Only records with one specific OrderId will match

  **muks1003** Highly Voted 2 years, 6 months ago

The given answers are correct, OrderId is the partition key and any query not involving the OrderId will be considered as Cross Partition (Or Fan-out) Query.

upvoted 20 times

  **[Removed]** Most Recent 10 months ago

the given answer is correct Y-Y-N

upvoted 2 times

  **azuredemo2022three** 1 year, 6 months ago

1. SELECT * FROM c WHERE c.orderDate = "2019-05-03"

Yes, this query will run as a cross-partition query. Since the query does not include the partition key (orderId), it needs to scan all partitions in the container to find matching documents.

2. SELECT * FROM c WHERE c.customerId = "5425"

No, this query will not run as a cross-partition query. The query includes the partition key (orderId) in the WHERE clause, allowing it to be routed to the specific partition containing documents for the given customerId.

3. SELECT * FROM c WHERE c.orderDate = "2019-05-03" and c.orderId = "5515d96e-5860-4fa0-a70c-c965e3ffeb49"

No, this query will not run as a cross-partition query. The query includes the partition key (orderId) in the WHERE clause, allowing it to be routed to the specific partition containing the document with the matching orderId and orderDate.

So, to summarize:

Yes, it will run as a cross-partition query.

No, it will not run as a cross-partition query.

No, it will not run as a cross-partition query.

upvoted 1 times

  **azuredemo2022three** 1 year, 6 months ago

My Mistake

Answer is

Yes, it will run as a cross-partition query.

Yes, it will not run as a cross-partition query.

No, it will not run as a cross-partition query.

upvoted 2 times

  **AndrySistiawan** 1 year, 5 months ago

for number 2, why answer is Yes? if it will not run as a cross-partition query. t thins if "it will not run as a cross-partition query" answer is NO,

upvoted 1 times

  **Torent2005** 2 years, 3 months ago

NNY

Cross-partition query means that your query hits only necessary data from logical partition based on partition key (query has filter eg. c.orderId). If not scan whole data. IF "orderId" is partition key then first two queries are not cross-partition queries.

<https://learn.microsoft.com/en-us/azure/cosmos-db/sql/how-to-query-container>



upvoted 2 times

  **codingdown** 2 years, 1 month ago

noooooe at all.

cross partition means it will scan more than 1 partition, easy as that

upvoted 4 times

  **lakime** 2 years, 8 months ago

As orderid is partition key, And if we will assume that this orderid will appear in different dates and in different customers - then those answers are correct

upvoted 4 times

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.

Suggested 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.

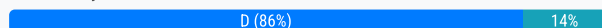
B: Sensor1Value has only two values.

C: All the devices could have the same manufacturer.

Reference:

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

Community vote distribution



muks1003 Highly Voted 2 years, 6 months ago

Selected Answer: D

D is the correct answer, I have taken the Microsoft Official Test And the answer is D.
upvoted 13 times

practicewizards Most Recent 2 weeks, 3 days ago

Selected Answer: D

it is correct
upvoted 1 times

bigleowski88 6 months, 2 weeks ago

Selected Answer: D



I believe there is a tradeoff between the huge fragmentation for D and the possibility of hotspots you get with the other options, Since this is a write heavy high volume application then the primary concern is reducing bottlenecks associated with hotspot partitions.
upvoted 2 times

[Removed] 10 months ago

Selected Answer: D

Answer is correct

upvoted 2 times

  **Gall** 2 years, 4 months ago

Selected Answer: D

Because of frequency (1s) we may not want to put too much data into the same container. The maxim quota for logical partition is 20GB, which can exhaust pretty fast.

upvoted 1 times

  **ognamala** 2 years, 4 months ago



Correct answer is D as per this link <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/synthetic-partition-keys> - in order to avoid a hot partition we have no option but to use a strategy like that suggested in the section "Use a partition key with a random suffix"

upvoted 2 times

  **chukismit** 2 years, 5 months ago



D, it's the right answer because deviceid a with pre-calculated suffixes number (Second when data is inserted) It will create 60 partitions by every deviceid

upvoted 3 times

  **jfarrell** 2 years, 6 months ago

I get the reason it is D BUT, there is context missing that would be necessary. For example, does this mean that the deviceid for like devices is the same and somehow hardware bound? Also, the suggestion of a random number is not inclusive of how random the number is, obviously if its generated with each send you will create excessive partitions



upvoted 2 times

  **lakime** 2 years, 7 months ago

Selected Answer: C

Fully agree, I would either select C, or have an answer - deviceid

upvoted 2 times

  **lakime** 2 years, 7 months ago



Sorry - I'm changing my mind ;) I guess B makes more sense in that case, we got two options (True/False) - and creating synthetic key when probably deviceid will have always same manufactor - doesn't make sense

upvoted 1 times

  **ognamala** 2 years, 4 months ago

C and B are both wrong, the both can end up creating a hot partition because of the low cardinality

upvoted 1 times

  **kdsingh** 2 years, 7 months ago

D, can't be the right answer because deviceid and a random number will create a new partition on every document you will write to the database.

upvoted 3 times

You maintain a relational database for a book publisher. The database contains the following tables.

Name	Column
Author	authorId (primary key)
	fullname
	address
	contactinfo
Book	bookId (primary key)
	isbn
	title
	genre
BookauthorInk	authorId (foreign key)
	bookId (foreign key)

The most common query lists the books for a given authorId.

You need to develop a non-relational data model for Azure Cosmos DB Core (SQL) API that will replace the relational database. The solution must minimize latency and read operation costs.

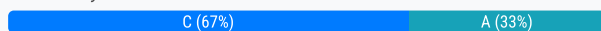
What should you include in the solution?

- A. Create a container for Author and a container for Book. In each Author document, embed bookId for each book by the author. In each Book document embed authorId of each author.
- B. Create Author, Book, and BookauthorInk documents in the same container.
- C. Create a container that contains a document for each Author and a document for each Book. In each Book document, embed authorId.
- D. Create a container for Author and a container for Book. In each Author document and Book document embed the data from BookauthorInk.

Suggested Answer: A

Store multiple entity types in the same container.

Community vote distribution



jfarrell Highly Voted 2 years, 6 months ago

This is nonsense. Microsoft's own documentation clearly states that for groups of data with like PKs (and it would be author here) use the same container. C is the answer

upvoted 19 times

grada Highly Voted 2 years, 5 months ago

Selected Answer: A

A is the closest of what I'd model, with a little add-on of both books and authors containing collections of not only the authorIds and bookIds (respectively), but also a subset of data from the documents they're referencing (for instance, authorId + authorName, and bookId + bookTitle).

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/modeling-data#hybrid-data-models>

upvoted 6 times

essdeecce 2 years, 3 months ago

Perfect. A link to the exact example in the documentation for this exact question. Not clear from the question but logically it [u]is[/u] a N:N relationship.

upvoted 4 times

practicewizards Most Recent 2 weeks, 3 days ago

Selected Answer: C

Rules of thumb, if have many entities, and need performance:

1 container and embed the others.

it is C.

upvoted 2 times

🗨️ 👤 **Blubb1860** 1 year ago

Selected Answer: C

C, is correct. You should not have two separate containers and you want to embed author information in books

upvoted 3 times

🗨️ 👤 **xRiot007** 1 year ago

C - one container for Author and Book and a prop in Book to link the author. An even better solution would be to add directly to the author the list of books, because the most used query is the list of books by author.

upvoted 1 times

🗨️ 👤 **rakun** 7 months, 4 weeks ago

"add directly to the author the list of books" - you are assuming stuff here, which is not good. Theoretically, if the author have 1 mil of books, you can create partition distribution skew, hot partition, hit limits and so on

upvoted 1 times

🗨️ 👤 **[Removed]** 1 year, 2 months ago

Correct Answer is A

upvoted 1 times

🗨️ 👤 **azuredemo2022three** 1 year, 6 months ago

Selected Answer: C

Selected Answer: C

upvoted 2 times

🗨️ 👤 **imando** 1 year, 9 months ago

Selected Answer: C

C is correct answer

upvoted 3 times

🗨️ 👤 **Ranzzzan** 1 year, 10 months ago

A

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/modeling-data#hybrid-data-models>

upvoted 3 times

🗨️ 👤 **monniq** 1 year, 3 months ago

A suggests for having 2 separate containers. C is in fact the Hybrid data model.

upvoted 1 times

🗨️ 👤 **NickKazen** 2 years, 2 months ago

Selected Answer: A

Although it would minimise read operation costs, there is no limit on the number of books authors can write (unfortunately in some cases). Because of that, the size of the actual document can grow enormously. That on its own will cause a huge factor of latency.

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/modeling-data#when-not-to-embed>

upvoted 1 times

🗨️ 👤 **codingdown** 2 years, 1 month ago

Then what?

If the number of books becomes very high, retrieving all the books per Author would require a high effort in both scenario (A or C) . Having the Author document embedded in the same container would not cause substantial overhead, (only extra operation to do in C would be to filter out the Author document based on property "type" AFTER retrieving all books (N)/ and author (1) documents having same partition key.

upvoted 1 times

🗨️ 👤 **DirectX** 2 years, 2 months ago

"there is no limit on the number of books authors can write (unfortunately in some cases)".

I think that this is not the case in real life. The ration is 1:few



upvoted 2 times

🗨️ 👤 **remz** 2 years, 4 months ago

Selected Answer: C

C is the most read cost effective

upvoted 2 times

  **remz** 2 years, 4 months ago

i read it back , A is the Right answer!

upvoted 1 times

  **ognamala** 2 years, 4 months ago



Selected Answer: C

We are explicitly required to minimise cost and latency, A although it would get the job done it would be highly inefficient - the relationship between an author and books is a one to few (1-many but bounded), hence data should be embedded - refer to the below links

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/modeling-data>

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-model-partition-example>

upvoted 5 times

  **jfarrell** 2 years, 6 months ago

Sorry I mean't B with a type field

upvoted 1 times

  **shachar_ash** 2 years, 6 months ago

In C you put them in the same container, and embedding to lower the latency

upvoted 2 times

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}]

Suggested Answer: A

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>

Community vote distribution

A (100%)

 **azuredemo2022three** Highly Voted 1 year, 6 months ago

Selected Answer: A

Selected Answer: A

upvoted 5 times

 **practicewizards** Most Recent 2 weeks, 3 days ago

Selected Answer: A

it is the other that has false in a string compare of IS_NUMBER

upvoted 1 times

 **Blubb1860** 1 year ago

Selected Answer: A

A, I tested it

upvoted 2 times

 **azuredemo2022three** 1 year, 6 months ago

Selected Answer: A

upvoted 1 times

 **Juba1711** 2 years, 1 month ago

correct

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/query/is-number>

upvoted 3 times

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.

Suggested Answer: C

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

F: 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>

Community vote distribution

C (100%)

 **kdsingh** Highly Voted 2 years, 7 months ago

C and E are correct answers.

upvoted 23 times

 **prim** Highly Voted 2 years, 5 months ago

C and E, in exam


upvoted 8 times

 **practicewizards** Most Recent 2 weeks, 3 days ago

Selected Answer: C

the trigger must be registered and pre-triggered.

upvoted 1 times

 **b01d700** 2 months, 4 weeks ago

Selected Answer: C

C and E

upvoted 1 times

 **Blubb1860** 1 year ago

Selected Answer: C

C and E are correct answers.

upvoted 1 times

 **SQLK** 2 years, 4 months ago


Is there an option F here

upvoted 3 times

 **susejzpol** 2 years, 2 months ago


I think that they refer to E when they are mentioning F option.

upvoted 1 times

 **Aunehwet79** 2 years, 4 months ago

Supposed to say E

upvoted 1 times

 **jfarrell** 2 years, 6 months ago

There is no F option

upvoted 4 times

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	

Answer Area

RequestOptions property to set:

	▼
AccessCondition	
ConsistencyLevel	
SessionToken	

Suggested Answer:

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>

 **AT96**  2 years, 1 month ago

When we then want to send our request to replace a document, we can specify an AccessCondition with the ETag we received when we fetched out our document.

upvoted 23 times

  **grada** 1 year, 11 months ago

This, it's supposed to be either `AccessCondition` or `IfMatchETag` directly, consistency levels have nothing to do with optimistic concurrency.

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.documents.client.requestoptions.accesscondition?view=azure-dotnet#examples>
upvoted 3 times

  **b890yc**  9 months ago

The namespace `Microsoft.Azure.Documents.Client` is deprecated. So the below link refers to old version of programming API. In other words, using `AccessCondition` is deprecated.

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.documents.client.requestoptions.accesscondition?view=azure-dotnet#examples>



The latest namespace is `Microsoft.Azure.Cosmos` and refer to the following links.

<https://learn.microsoft.com/en-us/dotnet/api/microsoft.azure.cosmos.container.replaceitemasync>

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

So the answer will be `ConsistencyLevel` and `_etag`.

upvoted 2 times

  **b890yc** 9 months ago

Apologies, so the answer should be `ConsistencyLevel` and `_id`.



Consistency level can be mentioned in `ItemRequestOptions` and `_id` should be passed to `Container.ReplaceItemAsync<T>` Method.

upvoted 1 times

  **Garyn** 9 months ago

Please correct the answer. it should `AccessCondition`, and `etag`.

upvoted 3 times

  **remz** 1 year, 10 months ago

`Accesscondition` `ETAG`

upvoted 3 times

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:

- ⇒ Users can submit an unlimited number of posts.
- ⇒ The average number of posts submitted by a user will be more than 1,000.
- ⇒ Posts can have an unlimited number of comments from different users.
- ⇒ The average number of comments per post will be 100, but many posts will exceed 1,000 comments.
- ⇒ 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"/>

Suggested 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.

  **avijitd** Highly Voted 2 years, 11 months ago

correct

upvoted 11 times

  **practicewizards** Most Recent 2 weeks, 3 days ago

correct: Y Y N

upvoted 1 times

  **hnkira** 7 months, 3 weeks ago

should be YYY, as the size of the document increases cost might increase, 1RU is the cost to read a document of 1KB, though with 20 interests stored as an array in the document we might not add more than 1KB but there is no constraint mentioned about the interest property (is it an object, text, what size is the limit)

upvoted 1 times

  **codingdown** 2 years, 7 months ago

i don't see why the write cost should increase, the read cost does

upvoted 2 times

  **susejzpol** 2 years, 7 months ago

i think could be because the operation need to do and upsert instead of an insert.

upvoted 3 times

  **muks1003** 3 years ago

Correct Answer Yes, Yes, No

upvoted 3 times

  **shachar_ash** 3 years ago

Correct

upvoted 1 times

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



Suggested 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



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>

remz Highly Voted 1 year, 4 months ago

Correct

upvoted 9 times

practicewizards Most Recent 2 weeks, 3 days ago

It is correct:

- Configure MaxItemCount.
- Run the query and provide...
- Append the results.

upvoted 1 times

[Removed] 8 months ago

Correct. Yes, you should configure the MaxItemCount property to limit the number of items returned per page to 80. Additionally, you should use the continuation token returned in the response to retrieve the next page of results.

upvoted 3 times

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 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

Suggested 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>

Community vote distribution

A (100%)

🗳️ 👤 **[Removed]** 8 months ago

Selected Answer: A

An Azure Monitor alert can be configured to trigger a function when certain conditions are met, such as the normalized request units per seconds.
upvoted 4 times

🗳️ 👤 **Amokrane** 12 months ago

Selected Answer: A
upvoted 1 times

🗳️ 👤 **ognamala** 1 year, 4 months ago

Selected Answer: A

Agreed
upvoted 4 times

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 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 the function to have an Azure CosmosDB trigger.

Does this meet the goal?

- A. Yes
- B. No

Suggested Answer: B

Instead configure an Azure Monitor alert to trigger the function.


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

Reference:

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

Community vote distribution

B (100%)

  **imando** 9 months ago

Selected Answer: B

No Azure Monitor is correct option. B is the correct answer.

upvoted 3 times

  **nitohime** 1 year ago

B!!!!!!!!!!!!!!

upvoted 1 times

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 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 application to use the change feed processor to read the change feed and you configure the application to trigger the function.

Does this meet the goal?

A. Yes

B. No

Suggested Answer: B

Instead configure an Azure Monitor alert to trigger the function.



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

Reference:

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

Community vote distribution

B (100%)

  **Tuopikson** 5 months, 3 weeks ago

Selected Answer: B

is correct answer. An Azure Monitor alert can be configured to trigger a function when certain conditions are met, such as the normalized request units per seconds, not change feed processor.

upvoted 1 times

  **imando** 9 months ago

Selected Answer: B

B is correct answer

upvoted 3 times

You have a database named db1 in an Azure Cosmos DB Core (SQL) API account.

You are designing an application that will use db1.

In db1, you are creating a new container named coll1 that will store online orders.

The following is a sample of a document that will be stored in coll1.

```
{
  "customerId" : "bba6fe24-6d97-4935-8d58-36baa4b8a0e1",
  "orderId" : "9d7816e6-f401-42ba-ad05-0e03de35c0b8",
  "orderDate" : "2021-09-29",
  "orderDetails" : []
}
```

The application will have the following characteristics:

- ⇒ New orders will be created frequently by different customers.
- ⇒ Customers will often view their past order history.

You need to select the partition key value for coll1 to support the application. The solution must minimize costs.

To what should you set the partition key?

- A. orderId
- B. customerId
- C. orderDate
- D. id

Suggested Answer: B

If most of your workload's requests are queries and most of your queries have an equality filter on the same property, this property can be a good partition key choice. For example, if you frequently run a query that filters on UserID, then selecting UserID as the partition key would reduce the number of cross-partition queries.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/partitioning-overview>

Community vote distribution


B (100%)

  **azuredemo2022three** 1 year ago

Selected Answer: B

Answer is B

upvoted 4 times

  **TRUESON** 1 year, 2 months ago

all orders from one customer will be stored on the same partition which will be beneficial for reads

upvoted 3 times

You have a container in an Azure Cosmos DB Core (SQL) API account that stores data about orders.

The following is a sample of an order document.

```
{
  "orderId" : "d4a9179b-5ead-43a3-b851-add9a71ac4b6",
  "customerId" : "f6e39103-bdc7-4346-9cfb-45daa4b2becf",
  "orderDate" : "2021-09-29",
  "orderItems" : [...],
  "total" : 12345
}
```

Documents are up to 2 KB.

You plan to receive one million orders daily.

Customers will frequently view their past order history.

You are evaluating whether to use orderDate as the partition key.

What are two effects of using orderDate as the partition key? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. You will exceed the maximum number of partition key values
- B. You will exceed the maximum storage per partition
- C. There will always be a hot partition
- D. Queries will run cross-partition

Suggested Answer: CD

Not C: But the problem is that when the application writes new data, the writes will always be directed to the same partition, based on whatever day it is. This results in what's called a hot partition, where we have a bottleneck that's going to quickly consume a great deal more of the reserved throughput you've provisioned for the container. Specifically, Cosmos DB evenly distributes your provisioned throughput across all the physical partitions in the container.

D: Customers will frequently view their past order history. OrderID will be used. Queries will run cross-partition

Incorrect:


Not B: 1 million a day x 2KB -> 2 GB of data/day while maximum storage across all items per (logical) partition: 20 GB

Reference:

<https://www.tallan.com/blog/2019/04/30/horizontal-partitioning-in-azure-cosmos-db> <https://docs.microsoft.com/en-us/azure/cosmos-db/concepts-limits>

Community vote distribution

CD (100%)

 **Ranzzan** Highly Voted 1 year, 10 months ago


correct.

Not A: There is no limit to the total number of physical partitions in your container. As your provisioned throughput or data size grows, Azure Cosmos DB will automatically create new physical partitions by splitting existing ones

Not B: as max is 20 GB.

So C and D correct answer

upvoted 5 times

 **binglebowski88** Most Recent 6 months, 2 weeks ago

Selected Answer: CD

C & D. The assumption is that the queries will mostly be getting recent orders or today's date orders which means that the logical partition containing all of today's orders will be overutilized or 'hot'

upvoted 1 times

 **xRiot007** 1 year ago

Answer is good explanation is bad.

D - queries will run cross-partition because customerId (not orderId) will be used by customers to list their order history, meaning that all order for a given customerId will be selected.

upvoted 1 times

  **[Removed]** 1 year, 8 months ago

Selected Answer: CD

Correct

upvoted 3 times

You have a container in an Azure Cosmos DB Core (SQL) API account. The container stores data about families. Data about parents, children, and pets are stored as separate documents.

Each document contains the address of each family. Members of the same family share the same partition key named familyId.

You need to update the address for each member of the same family that share the same address. The solution must meet the following requirements:

- ⇒ Be atomic, consistent, isolated, and durable (ACID).
- ⇒ Provide the lowest latency.

What should you do?

- A. Update the document of each family member separately by using a patch operation
- B. Update the document of each family member separately and set the consistency level to strong
- C. Update the document of each family member by using a transactional batch operation

Suggested Answer: C

Each transaction provides ACID (Atomicity, Consistency, Isolation, Durability) property guarantees.

Transactional batch operations offer reduced latency on equivalent operations.

Note: Transactional batch describes a group of point operations that need to either succeed or fail together with the same partition key in a container. In the .NET

SDK, the TransactionalBatch class is used to define this batch of operations. If all operations succeed in the order they are described within the transactional batch operation, the transaction will be committed. However, if any operation fails, the entire transaction is rolled back.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/transactional-batch>

Community vote distribution

C (100%)

🗳️ **imando** Highly Voted 1 year, 3 months ago

Selected Answer: C

Keyword is transactional

upvoted 6 times

🗳️ **binglebowski88** Most Recent 6 months, 2 weeks ago

Selected Answer: C

C. ACID means that if one operations fails in a group then they all fail together. the transactional batch operation offers this functionality.

upvoted 1 times

🗳️ **azuredemo2022three** 12 months ago

Selected Answer: C

C. Update the document of each family member by using a transactional batch operation

upvoted 2 times

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. Create a new synthetic key that contains deviceId and timestamp
- B. Use timestamp as the partition key
- C. Use deviceManufacturer as the partition key
- D. Use sensor1Value as the partition key

Suggested Answer: A

Concatenate multiple properties of an item.

You can form a partition key by concatenating multiple property values into a single artificial partitionKey property. These keys are referred to as synthetic keys.

For example, consider the following example document:

```
{
  "deviceId": "abc-123",
  "date": 2018
}
```

For the previous document, one option is to set /deviceId or /date as the partition key. Use this option, if you want to partition your container based on either device

ID or date. Another option is to concatenate these two values into a synthetic partitionKey property that's used as the partition key.

```
{
  "deviceId": "abc-123",
  "date": 2018,
  "partitionKey": "abc-123-2018"
}
```

Incorrect:

Not B: But the problem is that when the application writes new data, the writes will always be directed to the same partition, based on whatever day it is. This results in what's called a hot partition, where we have a bottleneck that's going to quickly consume a great deal more of the reserved throughput you've provisioned for the container. Specifically, Cosmos DB evenly distributes your provisioned throughput across all the physical partitions in the container.

Reference:

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

Community vote distribution

A (100%)

Selected Answer: A

A is correct answer. Device id + timestamp avoid hot partitions, minimize partition skew and capacity limits.

Choosing timestamp alone or static values like deviceManufacturer or sensor1Value wouldn't provide the dynamic distribution needed to handle frequent writes effectively. Thus, a combination key such as deviceId and timestamp is the most efficient way to achieve balanced partitioning for IoT device data.

upvoted 1 times

  **imando** 9 months ago

Selected Answer: A

Correct answer

upvoted 4 times

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:

- ⇒ Store companies and the users employed by them. Each company will have less than 1,000 users.
- ⇒ Some users have data that is greater than 2 KB.
- ⇒ Associate each user to only one company.
- ⇒ Provide the ability to browse by company.
- ⇒ Provide the ability to browse the users by company.
- ⇒ Whenever a company or user profile is selected, show a details page for the company and all the related users.
- ⇒ 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 the company ID as the partition key.
- B. Create a user container that uses the user ID as the partition key and a company container that uses the company ID as the partition key. Add the company ID to each user document.
- C. 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.
- D. In a company container, create a document for each company. Embed the users into company documents. Use the company ID as the partition key.

Suggested Answer: D

All employees within a company would nicely fit within a single document (document size 2 MB).

Note: An Azure Cosmos container is the unit of scalability both for provisioned throughput and storage. A container is horizontally partitioned and then replicated across multiple regions. The items that you add to the container are automatically grouped into logical partitions, which are distributed across physical partitions, based on the partition key.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/set-throughput>

Community vote distribution



Zyo Highly Voted 1 year, 5 months ago

1000 users with more than 2kb data can still exceed the 2MB allowed document size (dont forgot the company itself has data), so D is obviously incorrect.

upvoted 13 times

arnabdt Highly Voted 1 year, 9 months ago

Selected Answer: B

B. Create a user container that uses the user ID as the partition key and a company container that uses the company ID as the partition key. Add the company ID to each user document.

This design separates the users and companies into two different containers, with each container optimized for querying their respective entities. The user container uses the user ID as the partition key, which allows for efficient retrieval of individual users. The company container uses the company ID as the partition key, which enables efficient retrieval of all users associated with a particular company.

Adding the company ID to each user document in the user container allows for easy and fast querying of users by company. Whenever a company or user profile is selected, the details page for the company and all related users can be retrieved by joining the data from the two containers using the company ID.

This design meets all the requirements specified in the prompt, including efficient reading of data, browsing by company, and browsing users by company.

upvoted 13 times

xRiot007 1 year ago


The filter for both queries is company ID. B would result in cross partition queries with are not optimized for reading. A is the correct answer - one container, company ID as partition key and you can distinguish users from companies using a type prop, so if I only care to browse for a company I will not retrieve all its users because that would be extra data irrelevant for my search.

upvoted 8 times

 **WimTS** Most Recent 2 months, 2 weeks ago

Selected Answer: A

1 container and companyID as partitionKey to avoid cross-partition queries, and separate documents to avoid hitting the 2Mb limit (1000 users * 2KB)
upvoted 2 times

 **matejka** 3 months, 1 week ago

Selected Answer: D

D. In a company container, create a document for each company. Embed the users into company documents. Use the company ID as the partition key.

Here's why:

Efficiency in Data Retrieval: By embedding user data directly into the company document, all relevant user details can be fetched in a single query using the company ID. This aligns with the requirement to display a company's details and all related users on the details page.

Optimized for Read Operations: Embedding ensures that browsing by company or browsing users within a company doesn't require multiple queries or complex joins. This makes the design highly performant for read-heavy operations.

Scalability: Since each company has less than 1,000 users, embedding users into a company document won't exceed size or scalability limits of most document databases.

This approach is particularly effective when the use case prioritizes read performance and browsing-related data access patterns.

upvoted 1 times


 **szkielet** 4 months, 2 weeks ago

Selected Answer: D

Somebody has mention the limit.. But the limit of the document is up to 2MB not 2KB

<https://learn.microsoft.com/en-us/azure/cosmos-db/concepts-limits>

upvoted 1 times

 **szkielet** 4 months, 4 weeks ago

Selected Answer: A

definitely A

upvoted 1 times

 **biglebowski88** 6 months, 2 weeks ago

Selected Answer: A

I've changed my mind to A. A would not have the risk of hitting the 2mb size limit whilst also maintaining in partition queries based on the companyID partition key.

upvoted 1 times

 **biglebowski88** 6 months, 2 weeks ago

Selected Answer: D

D. Last statement says it should be optimized for reads. reads with embedded documents is most efficient and less compute resource intensive.

upvoted 2 times

 **[Removed]** 10 months ago

Selected Answer: B

option B is the most suitable design to optimize the data store for reading data

Partitioning data based on the company ID for companies and the user ID for users allows for efficient retrieval of data. When querying for a specific company or user, the partition key helps direct the query to the appropriate partition, minimizing the amount of data that needs to be scanned.

upvoted 1 times

 **monniq** 1 year, 3 months ago

Selected Answer: A

A is correct, since the company and users will be in the same partition and can be retrieved in an optimized way. You can use type property to query for the companies only, or users only.

B is incorrect since users are partitioned by userid and when retrieving users for a company it will be a cross-partition query (not optimal way)

D will create large company documents and it is not an optimal way again.

C again is not an optimal way since the query to get user by company will run cross partition.

upvoted 7 times

 **Womapoc** 1 year, 4 months ago

Selected Answer: D

D is correct

upvoted 2 times

🗨️ 👤 **akash211** 1 year, 5 months ago

B seems to be more optimized for reading data

upvoted 2 times

🗨️ 👤 **vinayhat** 1 year, 5 months ago

none of the options address "Associate each user to only one company". Ideally there should be a hierarchical key of userid / companyid to meet this requirement

upvoted 1 times

🗨️ 👤 **wojtekc** 1 year, 5 months ago

Selected Answer: D

Option D provides a well-optimized data store design that enables efficient browsing and retrieval of data by company, as well as an easy way to view company details and associated users on a single page.

upvoted 8 times

🗨️ 👤 **TRUESON** 1 year, 8 months ago

A is correct ... We want users and companies on the same partition. This is done by using company id as partition key for both. As a companyid is a widely used filter the data that are related can be covered from the same partition. we add a type pro to distinguish the two types

upvoted 8 times

🗨️ 👤 **imando** 1 year, 9 months ago

Selected Answer: D

D is correct answer. Googled other sites and they are showing D as answer.

upvoted 6 times

You are building an application that will store data in an Azure Cosmos DB Core (SQL) API account. The account uses the session default consistency level. The account is used by five other applications. The account has a single read-write region and 10 additional read region. Approximately 20 percent of the items stored in the account are updated hourly.

Several users will access the new application from multiple devices.

You need to ensure that the users see the same item values consistently when they browse from the different devices. The solution must NOT affect the other applications.

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

NOTE: Each correct selection is worth one point.

- A. Set the default consistency level to eventual
- B. Associate a session token to the device
- C. Use implicit session management when performing read requests
- D. Provide a stored session token when performing read requests
- E. Associate a session token to the user account

Suggested Answer: DE

Use Session consistency and use Stateful Entities in Durable Functions or something similar that will allow you to implement a distributed mutex to store and update the Session token across multiple Cosmos client instances.

Note: The session consistency is the default consistency that you get while configuring the cosmos DB account. This level of consistency honors the client session. It ensures a strong consistency for an application session with the same session token. What that means is that whatever is written by a session will return the latest version for reads as well, from that same session.

Incorrect:

Not A: Eventual consistency is the weakest consistency level of all. The first thing to consider in this model is that there is no guarantee on the order of the data and also no guarantee of how long the data can take to replicate. As the name suggests, the reads are consistent, but eventually.

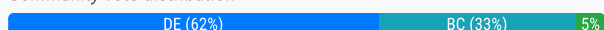
This model offers high availability and low latency along with the highest throughput of all. This model suits the application that does not require any ordering guarantee. The best usage of this type of model would be the count of retweets, likes, non-threaded comments where the count is more important than any other information.

Not B: The reads should be the same from different devices.

Reference:

<https://stackoverflow.com/questions/64084499/can-i-use-a-client-constructed-session-token-for-cosmosdb>

Community vote distribution



comoon Highly Voted 10 months, 1 week ago

Selected Answer: DE

In such case user would see the same data from the different device, because session would be the same
upvoted 9 times

imando Highly Voted 1 year, 1 month ago

Selected Answer: DE

D and E are correct answers
upvoted 6 times

Garyn Most Recent 9 months ago

BC is the right answer. Please update.

To ensure that users see the same item values consistently when browsing from different devices while not affecting the other applications, you should perform the following two actions:

B. Associate a session token to the device:

You should associate a session token with each user's device. This means that each device used by a user will have its own session token. This ensures that the same user, when browsing from different devices, maintains a consistent session with the Azure Cosmos DB. It allows users to see consistent item values across their devices.

C. Use implicit session management when performing read requests:

When performing read requests from the new application, use implicit session management. Implicit session management means that the SDK you are using to interact with Azure Cosmos DB automatically manages the session token for you. It keeps track of the session token associated with the device, ensuring that each read operation uses the correct session token. This ensures consistency for the user's session across devices.

Here's an additional explanation of the options you should not select:

upvoted 1 times

🗲️ 👤 **AshishKu** 9 months, 1 week ago

D and E are correct. We login to device with a user-id so session token should be associated with user not device.

upvoted 2 times

🗲️ 👤 **krcmyoyo** 10 months, 1 week ago

Selected Answer: BC

B. By associating a session token with each device, you ensure that reads from the same device maintain consistent and up-to-date results.

C. Implicit session management involves utilizing Cosmos DB's built-in session handling capabilities.

upvoted 2 times

🗲️ 👤 **azuredemo2022three** 1 year ago

Selected Answer: BC

Selected Answer: BC

upvoted 2 times

🗲️ 👤 **azuredemo2022three** 1 year ago

B. Associate a session token to the device.

C. Use implicit session management when performing read requests.

upvoted 2 times

🗲️ 👤 **aliounegdiop** 1 year ago

B. Associate a session token to the device.

E. Associate a session token to the user account.

By associating a session token to each device and user account, you can maintain consistency across multiple devices used by the same user. The session token represents a logical session and helps track the user's interaction with the data. This ensures that when the user switches between devices, they continue to see the same consistent data.

upvoted 1 times

🗲️ 👤 **[Removed]** 1 year, 2 months ago

Selected Answer: BD

By associating a session token to the device and providing a stored session token when performing read requests, the application ensures that the user sees the same item values consistently when browsing from different devices.

upvoted 1 times

🗲️ 👤 **arnabdt** 1 year, 3 months ago

Selected Answer: BC

B. Associate a session token to the device: When a user logs in to the application from a device, a session token should be generated and associated with the device. This will ensure that subsequent requests from the same device are routed to the same backend server that served the first request, and the user will see consistent item values.

C. Use implicit session management when performing read requests: When performing read requests, the application should use implicit session management. This means that the SDK will automatically manage the session token, and the user will continue to see consistent item values regardless of which backend server serves the request.

upvoted 3 times

🗲️ 👤 **arnabdt** 1 year, 3 months ago

It talk about different devices so I think I will go with D & E.

upvoted 4 times

HOTSPOT -

You have a container that stores data about families.
The following is a sample document.

```
{
  "lastName": "Cartwright",
  "parents": [
    {
      "firstName": "Elvira",
      "role": "mother",
      "age": 64
    },
    {
      "firstName": "Randolph",
      "role": "father",
      "age": 67
    }
  ],
  "children": [
    {
      "grade": 5,
      "pets": [

      ],
      "firstName": "Dana",
      "age": 15,
      "gender": "female"
    },
    {
      "grade": 7,
      "pets": [
        {
          "name": "Bob",
          "type": "guinea pig"
        }
      ],
    },
  ],
}
```

You run the following query against the container.

```
SELECT
  ch.name ?? ch.firstName AS childName,
  f.parents
  ARRAY_LENGTH(ch.pets) ?? 0 AS numberOfPets,
  ch.pets
FROM c AS f
JOIN ch IN f.children
```

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
Children who do not have parents defined will appear on the list	<input type="radio"/>	<input type="radio"/>
Children who have more than one pet will appear on the list multiple times	<input type="radio"/>	<input type="radio"/>
Children who do not have pets defined will appear on the list	<input type="radio"/>	<input type="radio"/>

Suggested Answer:

Answer Area

Statements	Yes	No
Children who do not have parents defined will appear on the list	<input type="radio"/>	<input checked="" type="radio"/>
Children who have more than one pet will appear on the list multiple times	<input type="radio"/>	<input checked="" type="radio"/>
Children who do not have pets defined will appear on the list	<input checked="" type="radio"/>	<input type="radio"/>

Box 1: No -

The result is empty, since the cross product of each item from source and an empty set is empty:

Note: Joins result in a complete cross product of the sets participating in the join. The result of an N-way join is a set of N-element tuples, where each value in the tuple is associated with the aliased set participating in the join and can be accessed by referencing that alias in other clauses.

Box 2: No -

The cross join is not on the pets.

Box 3: Yes -



The cross join is not on the pets.

Reference:



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

  **YJC** Highly Voted 2 years, 1 month ago



After I tested with JSON, the answer is Yes, No, Yes
upvoted 30 times

  **ferpin** 1 year, 9 months ago

You are right!
upvoted 1 times

  **susejzopol** 2 years, 1 month ago

You are right! In the first question the query return a value because "c" is represent the entire document and not only the parent's array.
upvoted 5 times


  **4415b99** 8 months, 3 weeks ago

it depends what is meant by `having a parent defined`. my understanding is that parent = document. so, it probably means that there is no document which corresponds to the format specified above and as a result those children have nowhere to exist.

so, the answer is No, No, Yes.

even if u put only { "children": [...] } as an item, it will mean that the parent is defined. while the parent's name, pets and other properties except children r not defined.

upvoted 1 times

  **basiltomato** Highly Voted 1 year, 10 months ago

Also tested this.

There's missing comma and should be ch.firstname instead of firstName:

```
SELECT
```

```
ch.name ?? ch.firstname AS childName, f.lastName, f.parents, ARRAY_LENGTH(ch.pets) ?? 0 AS numberOfPets,  
ch.pets
```

```
FROM mktest1 AS f
```

```
JOIN ch IN f.children
```

Answer is YES, NO, YES
upvoted 5 times

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 an Azure Cosmos DB Core (SQL) API account named account1 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 Azure Event Grid to send events to the function by using an Event Grid trigger in the function.

Does this meet the goal?

- A. Yes
- B. No

Suggested Answer: B

Instead configure an Azure Monitor alert to trigger the function.

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

Reference:

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

 **Ranzzan** Highly Voted 10 months ago

No, Use Azure monitor alert to trigger function using action group
upvoted 8 times

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

Suggested Answer: C

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>

Community vote distribution

C (100%)

 **lakime** Highly Voted 2 years, 8 months ago

Selected Answer: C

C is correct

<https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-configure-integrated-cache>

If you're using the .NET or Java SDK, set the connection mode to gateway mode. This step isn't necessary for the Python and Node.js SDKs since they don't have additional options of connecting besides gateway mo

upvoted 13 times

 **[Removed]** Most Recent 9 months, 4 weeks ago

Selected Answer: C

I can't tell if my application is using the dedicated gateway:

Check the `DedicatedGatewayRequests`. This metric includes all requests that use the dedicated gateway, regardless of whether they hit the integrated cache. If your application uses the standard gateway or direct mode with your original connection string, you will not see an error message, but the `DedicatedGatewayRequests` will be zero. If your application uses direct mode with your dedicated gateway connection string, you may still see a few `DedicatedGatewayRequests`.

<https://learn.microsoft.com/en-us/azure/cosmos-db/integrated-cache#i-cant-tell-if-my-application-is-using-the-dedicated-gateway>

upvoted 1 times

HOTSPOT -

You provision Azure resources by using the following Azure Resource Manager (ARM) template.

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "db": {
      "defaultValue": "[resourceId('Microsoft.DocumentDB/databaseAccounts', 'prod1')]",
      "type": "String"
    },
    "sms": {
      "defaultValue": "[resourceId('Microsoft.Insights/actionGroups', 'sms')]",
      "type": "String"
    }
  },
  "variables": {},
  "resources": [
    {
      "type": "microsoft.insights/actionGroups",
      "apiVersion": "2019-06-01",
      "name": "sms",
      "location": "Global",
      "properties": {
        "groupShortName": "Send message",
        "enabled": true,
        "emailReceivers": [],
        "smsReceivers": [
          {
            "name": "Action-SMS",
            "countryCode": "44",
            "phoneNumber": "7111111111"
          }
        ]
      }
    },
    {
      "type": "microsoft.insights/activityLogAlerts",
      "apiVersion": "2020-10-01",
      "name": "Alert1",
      "location": "Global",
      "dependsOn": ["sms"],
      "properties": {
        "scopes": [ "[parameters('db')]" ],
        "condition": {
          "allOf": [
            {
              "field": "category",
              "equals": "Administrative"
            },
            {
              "field": "operationName",
              "equals": "Microsoft.DocumentDB/databaseAccounts/regenerateKey/action"
            }
          ]
        },
        "actions": {
          "actionGroups": [
            {
              "actionGroupId": "[parameters('sms')]",
              "webhookProperties": {}
            }
          ]
        },
        "enabled": true
      }
    }
  ]
}
```

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
The alert will be triggered when an Azure Cosmos DB key is used	<input type="radio"/>	<input type="radio"/>
Two alert actions will be performed when the alert is triggered	<input type="radio"/>	<input type="radio"/>
The alert will be triggered when an item that has a new partition key value is created	<input type="radio"/>	<input type="radio"/>

Suggested Answer:

Answer Area

Statements	Yes	No
The alert will be triggered when an Azure Cosmos DB key is used	<input type="radio"/>	<input checked="" type="radio"/>
Two alert actions will be performed when the alert is triggered	<input type="radio"/>	<input checked="" type="radio"/>
The alert will be triggered when an item that has a new partition key value is created	<input checked="" type="radio"/>	<input type="radio"/>

Box 1: No -

An alert is triggered when the DB key is regenerated, not when it is used.

Note: The az cosmosdb keys regenerate command regenerates an access key for a Azure Cosmos DB database account.

Box 2: No -

Only an SMS action will be taken.


Emailreceivers is empty so no email action is taken.

Box 3: Yes -

Yes, an alert is triggered when the DB key is regenerated.

Reference:

<https://docs.microsoft.com/en-us/cli/azure/cosmosdb/keys>

  **Ritesh073** Highly Voted 1 year, 7 months ago

Answer should be No,No,No. Answer description is correct
upvoted 28 times

  **ognamala** Highly Voted 1 year, 4 months ago

The last one should also be no - the trigger is fired when the database account has a new key regenerated not when an item with a new partition key value is created
upvoted 9 times

  **imando** Most Recent 7 months ago

NO NO and NO
upvoted 4 times

  **ferpin** 9 months, 2 weeks ago

No, No, No. The third one is when a new access key is generated not when an item with a new partition key value is created.
upvoted 4 times

You plan to store order data in an Azure Cosmos DB Core (SQL) API account. The data contains information about orders and their associated items.

You need to develop a model that supports order read operations. The solution must minimize the number of requests.

What should you do?

- A. Create a database for orders and a database for order items.
- B. Create a single database that contains a container for orders and a container for order items.
- C. Create a single database that contains one container. Store orders and order items in separate documents in the container.
- D. Create a single database that contains one container. Create a separate document for each order and embed the order items into the order documents.

Suggested Answer: D

By denormalizing data, your application may need to issue fewer queries and updates to complete common operations.

Typically denormalized data models provide better read performance.

Note: In general, use embedded data models when:

There are contained relationships between entities.

There are one-to-few relationships between entities.

There's embedded data that changes infrequently.

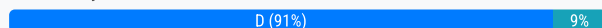
There's embedded data that will not grow without bound.

There's embedded data that is queried frequently together.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/modeling-data>

Community vote distribution



azuredemo2022three Highly Voted 1 year, 6 months ago

Selected Answer: D

Selected Answer: D

upvoted 7 times

rakun Most Recent 7 months, 3 weeks ago

See, in this kind of questions i think we must assume that the number of items per order can go to infinity. Thus, it would be C.

As answer description says:

Note: In general, use embedded data models when:

There's embedded data that will not grow without bound.

We do not see any bound information in this example and we do not know how much data goes with each item - is it 10 key-value pairs? is it 100? 1000? So this can cause our app to hit limits and requires complete re-design and migration.

upvoted 1 times

D3D1997 1 year, 6 months ago

Selected Answer: D

The solution must minimize the number of requests

upvoted 2 times

azuredemo2022three 1 year, 6 months ago

Selected Answer: B

upvoted 2 times

imando 1 year, 7 months ago

Selected Answer: D

D is answer

upvoted 2 times

arnabdt 1 year, 9 months ago

Selected Answer: B

B. Create a single database that contains a container for orders and a container for order items.

Creating a single database with two containers, one for orders and another for order items, is the best approach to support order read operations while minimizing the number of requests. This approach provides a clear separation of concerns between orders and order items, making it easier to manage the data and perform read operations.

With this approach, queries that require only order data can be directed to the orders container, while queries that require order item data can be directed to the order items container. This reduces the number of requests required to retrieve the data needed for each query.

upvoted 1 times

  **xRiot007** 1 year ago

You have order reads, not order item reads, so no need for 2 containers. To reduce number of requests you denormalize data by providing a document for each order and its items.

upvoted 1 times

You have a container in an Azure Cosmos DB Core (SQL) API account. The container stores telemetry data from IoT devices. The container uses telemetryId as the partition key and has a throughput of 1,000 request units per second (RU/s). Approximately 5,000 IoT devices submit data every five minutes by using the same telemetryId value.

You have an application that performs analytics on the data and frequently reads telemetry data for a single IoT device to perform trend analysis. The following is a sample of a document in the container.

```
{
  "id" : "9ccf1906-2a30-4dc0-9644-2185f5dcbbd7",
  "deviceId" : "bba6fe24-6d97-4935-8d58-36baa4b8a0e1",
  "telemetryId" : "9d7816e6-f401-42ba-ad05-0e03de35c0b8",
  "date" : "2019-05-03",
  "time" : "13:05",
  "temp" : "21"
}
```

You need to reduce the amount of request units (RUs) consumed by the analytics application.

What should you do?

- A. Decrease the offerThroughput value for the container.
- B. Increase the offerThroughput value for the container.
- C. Move the data to a new container that uses a partition key of deviceId.
- D. Move the data to a new container that uses a partition key of temp.

Suggested Answer: C

The analytics application would frequently read from the same partition of Device ID is used as partition key.

Note: The partition key determines how Azure Cosmos DB routes data in partitions. The IoT Device ID is the usual partition key for IoT applications.

Reference:

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

Community vote distribution

C (100%)

 **azuredemo2022three** Highly Voted 1 year ago

Selected Answer: C

Selected Answer: C

upvoted 5 times

 **Zaytsev** Most Recent 1 year, 2 months ago

Correct

upvoted 4 times

HOTSPOT

-

You plan to use a multi-region Azure Cosmos DB Core (SQL) API account to store data for a new application suite. The suite contains the applications shown in the following table.

Name	Requirement
Reporting	Must be able to track the total order counts within five minutes of orders being placed.
Purchasing	Must guarantee that the latest committed stock quantities are used always.
Fulfillment	Must ensure that orders are read in the order in which they are placed.

Each application should use the weakest consistency level possible.

Which consistency level should you configure for each application? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

Reporting: ▼

- Bounded staleness
- Consistent prefix
- Eventual
- Session
- Strong

Purchasing: ▼

- Bounded staleness
- Consistent prefix
- Eventual
- Session
- Strong

Fulfillment: ▼

- Bounded staleness
- Consistent prefix
- Eventual
- Session
- Strong

Answer Area

Suggested Answer:

Reporting:
Bounded staleness
Consistent prefix
Eventual
Session
Strong

Purchasing:
Bounded staleness
Consistent prefix
Eventual
Session
Strong

Fulfillment:
Bounded staleness
Consistent prefix
Eventual
Session
Strong

 **sc200** Highly Voted 1 year, 6 months ago

Reporting: Bounded staleness (within 5 mins)

Purchasing: Strong (latest committed)

Fulfillment: Consistent prefix (never out-of-order)

<https://willvelida.medium.com/understanding-consistency-levels-in-azure-cosmos-db-77b90dbf3daa>

upvoted 44 times

 **leoli9001** Most Recent 10 months, 2 weeks ago

Reporting: Bounded staleness (within 5 mins)

Purchasing: Strong (latest committed)

Fulfillment: Consistent prefix (never out-of-order)

<https://learn.microsoft.com/en-us/training/modules/use-consistency-models-azure-cosmos-db-sql-api/2-understand>

upvoted 3 times

 **comoon** 1 year, 4 months ago

Reporting: Eventual

Purchasing: Strong

Fulfillment: Consistent prefix

upvoted 2 times

 **azuredemo2022three** 1 year, 6 months ago

Reporting: Eventual

Purchasing: Bounded Staleness

Fulfillment: Strong

upvoted 1 times

HOTSPOT

-

You are designing a data model for an Azure Cosmos DB Core (SQL) API account.

What are the partition limits for request units per second (RU/s) and storage? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

Maximum RU/s per physical partition:

400
5,000
10,000
Unlimited

Maximum storage per logical partition:

10 GB
20 GB
50 GB
Unlimited

Answer Area

Suggested Answer:

Maximum RU/s per physical partition:

400
5,000
10,000
Unlimited

Maximum storage per logical partition:

10 GB
20 GB
50 GB
Unlimited

 **azuredemo2022three** Highly Voted 1 year ago

Maximum RU/s per physical partition: 10,000 RU/s

Maximum storage per logical partition: 20 GB

upvoted 35 times

 **whinycarebear** 11 months, 3 weeks ago

Correct

<https://learn.microsoft.com/en-us/azure/cosmos-db/concepts-limits>

upvoted 5 times

  **akash211** Most Recent 11 months ago

It is 20 GB. 50 GB is the maximum size of a physical partition.

upvoted 4 times

HOTSPOT

-

You have an Azure Cosmos DB container named container1.

You need to insert an item into container1. The solution must ensure that the item is deleted automatically after two hours.

How should you complete the item definition? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
{
  "id": "1",
  "customerOrderNumber": "XA2342093",
  "orderDate": "2022-04-22T09:34:42",
  "customerNumber": 234223,
  "orderItems":
  [
    {
      "id": 1,
      "quantity": 12,
      "productCode": "B234BLK"
    }
  ]
  "_attachments": "attachments/",
  "   ":  ,
  

defaultTimeToLive  
ttl  
_ttl


  "_ts": 1551322496
  

-1  
2  
120  
7200


}
```


Answer Area

```
{
  "id": "1",
  "customerOrderNumber": "XA2342093",
  "orderDate": "2022-04-22T09:34:42",
  "customerNumber": 234223,
  "orderItems":
  [
    {
      "id": 1,
      "quantity": 12,
      "productCode": "B234BLK"
    }
  ]
  "_attachments": "attachments/",
  " : ",
  defaultTimeToLive
  ttl
  _ttl
  "-ts": 1551322496
  7200
```

Suggested Answer:

 **azuredemo2022three** Highly Voted 1 year, 6 months ago

ttl and 7200

upvoted 14 times

 **xRiot007** Most Recent 1 year ago

Answer is ttl and 7200.

The default time to live is something you set on the container when using the SDK

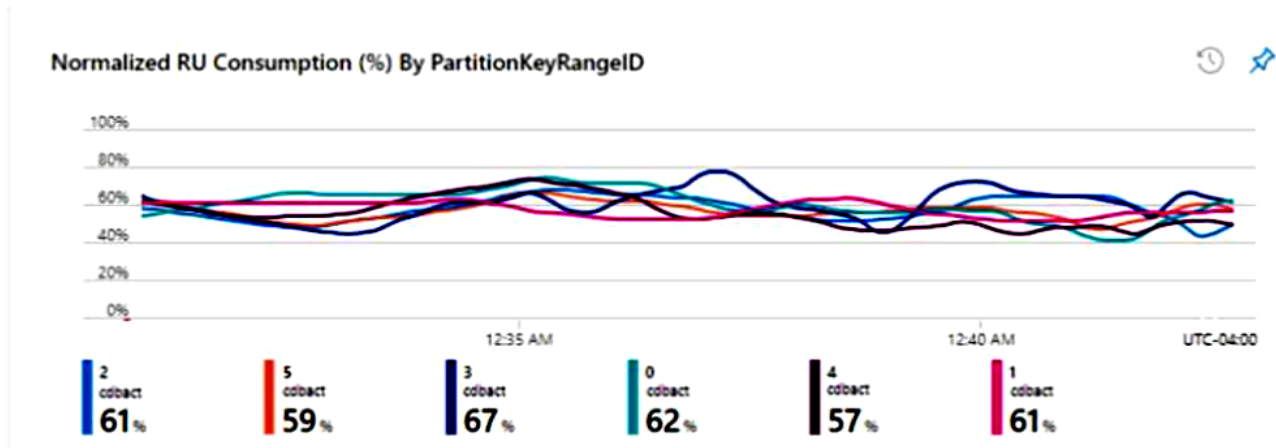
upvoted 2 times

HOTSPOT

-

You have a container in an Azure Cosmos DB Core (SQL) API account. The database that has a manual throughput of 30,000 request units per second (RU/s).

The current consumption details are shown in the following chart.



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

NOTE: Each correct selection is worth one point.

Answer Area

Each partition supports throughput of up to [answer choice] RU/s.

5,000
10,000
20,000
30,000

The container can scale to [answer choice] RU/s without a partition split.

10,000
20,000
30,000
60,000

Suggested Answer:


Answer Area

Each partition supports throughput of up to [answer choice] RU/s.

5,000
10,000
20,000
30,000

The container can scale to [answer choice] RU/s without a partition split.

10,000
20,000
30,000
60,000

 **noah1995** Highly Voted 1 year, 6 months ago

Answer 1: 5000 RU/s. You get this by dividing 30000 RU/s by 6 physical partitions.

Answer 2: 60000 RU/s. The highest RU/s you can scale to without triggering Azure Cosmos DB to split partitions is equal to Current number of physical partitions * 10,000 RU/s.

<https://learn.microsoft.com/en-us/azure/cosmos-db/scaling-provisioned-throughput-best-practices#step-2-calculate-the-default-maximum-throughput>

upvoted 17 times

 **azuredemo2022three** Highly Voted 1 year, 6 months ago

In an Azure Cosmos DB Core (SQL) API account, each partition supports throughput of up to 10,000 request units per second (RU/s).

The container can scale to a maximum throughput of 30,000 RU/s without requiring a partition split. This is because the maximum RU/s per physical partition in Azure Cosmos DB is 10,000, and since the container is set to a manual throughput of 30,000 RU/s, it can be accommodated within three physical partitions, each handling 10,000 RU/s.

Therefore, each partition supports up to 10,000 RU/s, and the container can scale up to 30,000 RU/s without a partition split.

upvoted 12 times

 **szkielet** 4 months, 4 weeks ago

wrong, it can not scale up because it is manually set as 30,000 so if somebody would like to scale it manually to meet the limit then the 60,000 is the correct one.

upvoted 1 times

 **WimTS** Most Recent 2 months, 2 weeks ago


<https://learn.microsoft.com/en-us/azure/cosmos-db/scaling-provisioned-throughput-best-practices>

There's an example there.

Answer 1: 5000 RU/s - throughput is evenly distributed between the 6 physical partitions


Answer 2: That's a tricky one. Since it's manual, it will not go beyond 30.000 RU/s. But I think they mean how high you can possibly scale it to, and that would be 60.000, because each partition maximum can use 10.000

upvoted 1 times

 **Garyn** 1 year, 3 months ago


The correct answer should be 5000 RU/s and 60000 RU/s

upvoted 1 times

 **Garyn** 1 year, 3 months ago

Correction. it is 10000 RU/s and 60000 RU/s

upvoted 5 times

 **NK203** 1 year, 3 months ago

How can you know there are 6 physical partitions?

upvoted 1 times

 **Melon12** 11 months, 1 week ago

- Each partition (logical/physical) support upto 10000RU/s

- Graph shows based on partition key id. There are 6 partition key legend. So, The highest RU/s you can scale to without triggering Azure Cosmos DB to split partitions is equal to Current number of physical partitions * 10,000 RU/s. So, $6 \times 10000 \text{RU/s} = 60000 \text{RU/s}$

ref: <https://learn.microsoft.com/en-us/azure/cosmos-db/scaling-provisioned-throughput-best-practices>

upvoted 3 times

  **noah1995** 1 year, 6 months ago

5000 RU/s

10000 RU/s

upvoted 1 times

  **noah1995** 1 year, 6 months ago

Excuse me, I meant 5000 RU/s and 30000 RU/s



upvoted 2 times

  **XiangRongChang** 1 year, 5 months ago

5000 RU/s and 30000 RU/s or 5000 RU/s. and 60000 RU/s.

You give several answers.

upvoted 1 times

  **harass** 1 year, 3 months ago

The RU for the database is fixed at 30,000, so it cannot exceed 30,000.

The correct answers should be 5,000 RU and 30,000 RU.

upvoted 2 times

You have an Azure Cosmos DB database.

You plan to create a new container named container1 that will store product data and product category data and will primarily support read requests.

You need to configure a partition key for container1. The solution must meet the following requirements:

- Minimize the size of the partition.
- Minimize maintenance effort.

Which two characteristics should you prioritize? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. unique
- B. high cardinality
- C. low cardinality
- D. static

Suggested Answer: BD

Community vote distribution

BD (100%)

 **azuredemo2022three**  1 year, 6 months ago

Selected Answer: BD

To configure a partition key for container1 in Azure Cosmos DB, prioritizing characteristics that minimize the size of the partition and maintenance effort, you should prioritize the following two characteristics:

B. High cardinality: Choosing a partition key with high cardinality means having a large number of unique values distributed across the dataset. This helps in distributing the data evenly across multiple partitions, preventing hotspots and ensuring efficient scalability.

D. Static: Selecting a static partition key means choosing a key that does not frequently change. This minimizes the need for updates to the partition key value and reduces maintenance effort. It also helps in maintaining a consistent distribution of data across partitions over time.

So, the correct answers are:

- B. High cardinality
 - D. Static
- upvoted 11 times

 **xRiot007**  1 year ago

B,D - considering products have category data, we can expect the category to be a filter, which makes it a good candidate to be a partition key. Categories are static known terms for a given business (values like "Electronics", "Clothing", "Wellness" etc)

upvoted 1 times

 **Emil_Topics** 1 year, 4 months ago



A unique
B high cardinality
<https://medium.com/codex/azure-cosmos-db-partitioning-bc851a404476>
upvoted 3 times

 **Womapoc** 1 year, 4 months ago

Selected Answer: BD

correct

upvoted 1 times

  **TT924** 1 year, 4 months ago

A, B

<https://learn.microsoft.com/en-us/azure/cosmos-db/partitioning-overview#use-item-id-as-the-partition-key>

upvoted 4 times

HOTSPOT

-

You have an Azure Cosmos DB account named account1 that has a default consistency level of session.

You have an app named App1.

You need to ensure that the read operations of App1 can request either bounded staleness or consistent prefix consistency.

What should you modify for each consistency level? To answer, select the appropriate options in the answer area.

Answer Area

Bounded staleness:

The account level options
 The database level options
 The request level options

Consistent prefix:

The account level options
 The database level options
 The request level options



Answer Area**Suggested Answer:**

Bounded staleness:

The account level options
 The database level options
 The request level options


Consistent prefix:

The account level options
 The database level options
 The request level options


 **meiwein**  1 year, 5 months ago
account level & request level options

"Consistency can only be relaxed at the SDK instance or request level. To move from weaker to stronger consistency, update the default consistency for the Azure Cosmos DB account."

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/how-to-manage-consistency?tabs=portal%2Cdotnetv2%2Capi-async>
upvoted 16 times

 **bibleowski88** 6 months, 2 weeks ago
to clarify order in strength of consistency models is:
Strong (side note. not applicable to Multi Region Write architecture!),
Bounded Staleness,
Session,
Consistent Prefix
Eventual.

Bounded Staleness is stronger than session so cant be overridden in the SDK level.
Consistent Prefix is weaker than session so it can be overridden in the SDK level.
upvoted 2 times

 **[Removed]** 1 year, 4 months ago
Bounded staleness > account level
Consistent prefix > request level
"The default consistency level configured on your account applies to all Azure Cosmos DB databases and containers under that account."
<https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels#configure-the-default-consistency-level>
**No option to set consistency level at database level. requirements for multiple default consistency levels = multiple accounts
upvoted 3 times

🗨️ 👤 **xofowi5140** Most Recent 1 year, 5 months ago

"account level & request level options" or "The database level options & The request level options"?

upvoted 1 times

🗨️ 👤 **[Removed]** 9 months, 4 weeks ago

Consistency can only be relaxed at the SDK instance or request level. To move from weaker to stronger consistency, update the default consistency for the Azure Cosmos DB account.

For Bounded staleness --> Account

For Consistent prefix --> Request Level

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/how-to-manage-consistency?tabs=portal%2Cdotnetv2%2Capi-async#override-the-default-consistency-level>

upvoted 1 times

🗨️ 👤 **azuredemo2022three** 1 year, 6 months ago

For Bounded staleness:

The database level options

For Consistent prefix:

The request level options

upvoted 4 times

DRAG DROP

-

You are designing three apps named App1, App2, and App3. Each app will use a separate Azure Cosmos DB for NoSQL account. The apps have the following consistency requirements:

- App1: Reads must always return the most recent committed version of an item, where the commit occurred during the same session.
- App2: Reads must always return the most recent committed version of an item, even if the commit occurred during another session.
- App3: Write latency must be minimized and data staleness can be tolerated.

You need to recommend a default consistency level for each Azure Cosmos DB for NoSQL account. The solution must minimize concurrency.

What should you recommend for the account of each app? To answer, drag the appropriate consistency levels to the correct apps. Each consistency level may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Consistency levels**Answer Area**App1: App2: App3: **Answer Area****Suggested Answer:** App1: App2: App3:  **tdetry** Highly Voted 6 months, 3 weeks ago

correct

upvoted 6 times

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.

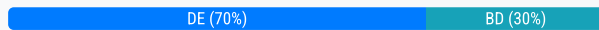
Which two factors can influence the SDK? Each correct answer presents a complete solution.

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

Suggested Answer: DE

Community vote distribution



[Removed] 9 months, 4 weeks ago

Selected Answer: DE

In order to take advantage of global distribution, client applications can specify the ordered preference list of regions to be used to perform document operations. Based on the Azure Cosmos DB account configuration, current regional availability and the preference list specified, the most optimal endpoint will be chosen by the SQL SDK to perform write and read operations.

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/tutorial-global-distribution?tabs=dotnetv2%2Capi-async#preferred-locations>
upvoted 4 times

SwePha 1 year, 1 month ago

Selected Answer: DE

Based on the Azure Cosmos DB account configuration, current regional availability and the preference list specified, the most optimal endpoint will be chosen by the SQL SDK to perform write and read operations.
upvoted 4 times

dkks 1 year, 1 month ago

Selected Answer: BD

answer

upvoted 4 times

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

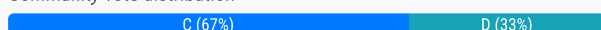
You need to create a container named Container1 in account1 by using the Azure Cosmos DB .NET SDK. The solution must ensure that the items in Container1 never expire.

What should you set?

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

Suggested Answer: D

Community vote distribution



bcd_6 1 year, 7 months ago

Selected Answer: C

Set to null will never expire.

If set to -1 might override by the item's ttl
upvoted 7 times

xRiot007 1 year, 6 months ago

When creating a container de DefaultTimeToLive has to be set to -1 for items to never expire. Overrides are not specified in this question and even if they were, this prop cannot be null, because the SDK is expecting a non-nullable integer.
upvoted 2 times

xRiot007 1 year, 6 months ago

Correction - it can be null, but in that case TTL is turned off. To have items that never expire you must set it to -1.
upvoted 1 times

xRiot007 1 year, 6 months ago

More details here: <https://learn.microsoft.com/en-us/dotnet/api/microsoft.azure.documents.documentcollection.defaulttimetolive?view=azure-dotnet>
upvoted 1 times

Blubb1860 1 year, 6 months ago

Selected Answer: C

Set to null will never expire.
upvoted 5 times

Examdumps2023 1 year ago

Selected Answer: D

To create a container named Container1 in account1 using the Azure Cosmos DB .NET SDK and ensure that the items in Container1 never expire, you should configure the DefaultTimeToLive property. Setting the DefaultTimeToLive to -1 ensures that items never expire.
upvoted 3 times

Bartus007 1 year, 2 months ago

ttl = -1 TTL is disabled. The item will never expire.
upvoted 1 times

[Removed] 1 year, 4 months ago

Selected Answer: D

Apologies, the correct choice is option C, setting DefaultTimeToLive to null. This ensures that items in Container1 will never expire.

If you set DefaultTimeToLive to null (option C), the container items will never expire, regardless of any TTL set for individual items.

Option D, setting DefaultTimeToLive to -1, is incorrect because it would prevent the container TTL from expiring, but if you set a TTL for individual items, those items would still expire based on their TTL value.

For more details, please refer to Example 1 and Example 2 in the following link: <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/time-to-live>

upvoted 1 times

🗲️ 👤 [Removed] 1 year, 4 months ago

Selected Answer: C

Apologies, the correct choice is option C, setting DefaultTimeToLive to null. This ensures that items in Container1 will never expire.

If you set DefaultTimeToLive to null (option C), the container items will never expire, regardless of any TTL set for individual items.

Option B, setting DefaultTimeToLive to -1, is incorrect because it would prevent the container TTL from expiring, but if you set a TTL for individual items, those items would still expire based on their TTL value.

For more details, please refer to Example 1 and Example 2 in the following link: <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/time-to-live>

upvoted 2 times

🗲️ 👤 [Removed] 1 year, 4 months ago

Selected Answer: D

TimeToLivePropertyPath specifies a path within the JSON document that indicates the expiration time. Setting it to null disables time-to-live (TTL) for individual items, while setting it to 0 would immediately expire the items. Both options are not suitable for ensuring items never expire. DefaultTimeToLive sets the default expiration time for all items in the container. Setting it to -1 signifies that the items will never expire. This is the desired behavior for this scenario.

upvoted 1 times

🗲️ 👤 [Removed] 1 year, 4 months ago

Apologies, the correct choice is option C, setting DefaultTimeToLive to null. This ensures that items in Container1 will never expire.

If you set DefaultTimeToLive to null (option C), the container items will never expire, regardless of any TTL set for individual items.

Option D, setting DefaultTimeToLive to -1, is incorrect because it would prevent the container TTL from expiring, but if you set a TTL for individual items, those items would still expire based on their TTL value.

For more details, please refer to Example 1 and Example 2 in the following link: <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/time-to-live>

upvoted 1 times

🗲️ 👤 [Removed] 1 year, 4 months ago

Apologies, the correct choice is option C, setting DefaultTimeToLive to null. This ensures that items in Container1 will never expire.

If you set DefaultTimeToLive to null (option C), the container items will never expire, regardless of any TTL set for individual items.

Option B, setting DefaultTimeToLive to -1, is incorrect because it would prevent the container TTL from expiring, but if you set a TTL for individual items, those items would still expire based on their TTL value.

For more details, please refer to Example 1 and Example 2 in the following link: <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/time-to-live>

upvoted 1 times

🗲️ 👤 SwePha 1 year, 7 months ago

Selected Answer: D

// Never expire by default

DefaultTimeToLive = -1

upvoted 2 times

You are designing an Azure Cosmos DB for NoSQL solution to store data from IoT devices. Writes from the devices will occur every second. Data will be retained indefinitely.

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 deviceManufacturer as the partition key.
- B. Create a new synthetic key that contains deviceId and timestamp.
- C. Create a new synthetic key that contains deviceId and deviceManufacturer.
- D. Use deviceId as the partition key.

Suggested Answer: B

Community vote distribution


B (100%)

 **Blubb1860** 1 year ago

Selected Answer: B

B is correct

upvoted 3 times

 **dkssk** 1 year, 1 month ago

Selected Answer: B

B selected

upvoted 2 times

HOTSPOT

-

You have an Azure Cosmos DB for NoSQL container named Contacts that is configured as shown in the following exhibit.

* Container id ⓘ

* Indexing

☒ Automatic ☐ Off

All properties in your documents will be indexed by default for flexible and efficient queries. [Learn more](#)

* Partition key ⓘ

Unique keys ⓘ



Contacts contains the items shown in the following table.

companyID	familyName	givenNames	emailAddress
1	Rogers	Terry John	terryj@contoso.com
1	Rogers	null	rogers@contoso.com

To Contacts, you plan to Insert the items shown in the following table.

Item	companyID	familyName	givenNames	emailAddress
1	1	Rogers	terry john	terryj@contoso.com
2	1	Rogers	null	rogers2@contoso.com
3	2	Rogers	Terry John	terryj@contoso.com

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Answer Area

Statements

Yes**No**

Item 1 can be inserted

☐☐

Item 2 can be inserted

☐☐

Item 3 can be inserted

☐☐

Answer Area

Statements

Yes

No

Suggested Answer:

Item 1 can be inserted

☒☐

Item 2 can be inserted

☐☒

Item 3 can be inserted

☒☐

  **SwePha** Highly Voted 1 year, 1 month ago

YNY



Unique Key is per partition and is case sensitive

upvoted 7 times

  **[Removed]** Most Recent 9 months, 3 weeks ago

YNY is the correct answer



upvoted 1 times

  **bcd_6** 1 year, 1 month ago

N N Y

There are two unique key rules



upvoted 1 times

  **bcd_6** 1 year, 1 month ago

Sorry, should be YNY.



After check the azure portal UI, I notice that is a composite contract with 2 fileds

upvoted 1 times

  **dkssks** 1 year, 1 month ago

NyN Selected

upvoted 1 times

  **dkssks** 1 year, 1 month ago

YNY selected

upvoted 2 times

HOTSPOT

-

You plan to create an Azure Cosmos DB container named account that will contain items in the following format.

```
{
  "id": "account-2022080712345",
  "balanceStatus": [
    {
      "balance": 1000,
      "timestamp": "2021-09-15T23:14:25.7251173Z"
    },
    {
      "balance": -10,
      "timestamp": "2021-09-11T19:16:25.1343234Z"
    },
    {
      "balance": 100,
      "timestamp": "2021-08-14T21:13:39.8121133Z"
    }
  ]
}
```

You need to define a query that will return the ids of all accounts that have ever recorded a negative value for balance.

How should you complete the query? To answer select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

SELECT c.id

FROM c

WHERE EXISTS(

SELECT n

FROM

	▼
c.balanceStatus c.balanceStatus.balance n IN c.balanceStatus	

WHERE

	▼
n < 0 n.balance n.balanceStatus balance	

< 0



)

Answer Area


```
SELECT c.id  
FROM c  
WHERE EXISTS(  
  SELECT n
```

Suggested Answer:

```
FROM  
  c.balanceStatus  
  c.balanceStatus.balance  
  n IN c.balanceStatus  
WHERE  
  n < 0  
  n.balance  
  n.balanceStatus balance  
)
```

  **[Removed]** 9 months, 4 weeks ago

Provided answer is correct
upvoted 2 times

  **SwePha** 1 year, 1 month ago

Answers - BalanceStatus and Balance
upvoted 2 times

HOTSPOT

-

You have the Azure Cosmos DB for NoSQL containers shown in the following table.

Name	DefaultTimeToLive
container1	-1
container2	null
container3	60

You have the items shown in the following table.

Name	Container	TimeToLive
item1	container1	60
item2	container2	10
item3	container3	-1

When will each item expire? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

Item1:

	▼
Never	
10 seconds	
60 seconds	
10 minutes	
60 minutes	

Item2:

	▼
Never	
10 seconds	
60 seconds	
10 minutes	
60 minutes	

Item3:

	▼
Never	
10 seconds	
60 seconds	
10 minutes	
60 minutes	


Answer Area

Item1:

Item2:

Item3:

Suggested Answer:

 **SwePha** Highly Voted 1 year, 1 month ago

60

Never

Never

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

If Null on container, TTL is disabled no matter what we set at item level, so it wont expire

upvoted 12 times

 **Blubb1860** 1 year ago

correct

upvoted 3 times

 **dkssk** Most Recent 1 year, 1 month ago

60

10

never

upvoted 4 times

Your company develops an application named App1 that uses the Azure Cosmos DB SDK and the Eventual consistency level.

App1 queries an Azure Cosmos DB for NoSQL account named account1.

You need to identify which consistency level to assign to App1 to meet the following requirements:

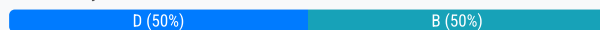
- Maximize the throughput of the queries generated by App1 without increasing the number of request units currently used by the queries.
- Provide the highest consistency guarantees.

Which consistency level should you identify?

- A. Strong
- B. Bounded Staleness
- C. Session
- D. Consistent Prefix

Suggested Answer: D

Community vote distribution



matejka 3 months, 2 weeks ago

Selected Answer: C

Throughput Optimization: Session consistency offers a balance between strong consistency and high throughput, which makes it a good option for applications that require predictable performance without significantly increasing the Request Unit (RU) cost.

Highest Consistency Guarantees: Among the weaker consistency levels (below Strong and Bounded Staleness), Session consistency provides strong consistency guarantees within the session context of a single user. It ensures monotonic reads, monotonic writes, and read-your-own-writes for requests within the same session.

While Strong (A) and Bounded Staleness (B) offer higher consistency guarantees than Session, they tend to use more RUs, potentially reducing throughput. Eventual (default) and Consistent Prefix (D) have higher throughput but offer weaker consistency guarantees compared to Session.

upvoted 1 times

szkielet 4 months, 4 weeks ago

Selected Answer: D

Consistent Prefix: Consistent prefix guarantees that all reads will see writes in the order they were made. It's the strongest consistency level that still allows for high throughput. It's stronger than session consistency because it guarantees global ordering of writes, not just within a session.

upvoted 1 times

binglebowski88 6 months, 2 weeks ago

Selected Answer: D

Consistent prefix will not increase the RU consumption whilst providing better consistency guarantee than session and eventual. the other two increase RU consumption and therefore incorrect.

upvoted 1 times

[Removed] 9 months, 4 weeks ago

Selected Answer: D

For strong and bounded staleness, reads are done against two replicas in a four replica set (minority quorum) to provide consistency guarantees. Session, consistent prefix and eventual do single replica reads. The result is that, for the same number of request units, read throughput for strong and bounded staleness is half of the other consistency levels.

<https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels#consistency-levels-and-throughput>

upvoted 2 times

[Removed] 8 months, 1 week ago

This is true, however, Session also does single replica reads and is stronger compared to consistent prefix, so why is it not chosen as an answer?
upvoted 3 times

🗨️ 👤 **biglebowski88** 6 months, 2 weeks ago

I think because session consistency only provides that consistency strength in the context of a user session so its a bit of an outlier. I dont think we can take that hierarchy at face value as it depends on the context. bit of a weird question without more context.
upvoted 1 times

🗨️ 👤 **biglebowski88** 6 months, 2 weeks ago

Session falls back to Eventual for data that is written by someone else apparently . so consistent prefix is definitely the best option for all round consistency and without impacting RUs
upvoted 1 times

🗨️ 👤 **xRiot007** 12 months ago

Answer is D - Consistent prefix.

Why ? We need to provide the highest consistency guarantees without decreasing throughput. Strong and bounded staleness provide half the throughput of Session, Consistent Prefix and Eventual, so by choosing them you decrease the throughput.
upvoted 3 times

🗨️ 👤 **xRiot007** 12 months ago

More details here: <https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels>
upvoted 2 times

🗨️ 👤 **xRiot007** 12 months ago

<https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels#consistency-levels-and-throughput>
upvoted 2 times

🗨️ 👤 **RedSquirrel** 1 year ago

Selected Answer: B

<https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels>
upvoted 3 times

🗨️ 👤 **tdetry** 1 year ago

highest consistency guarantees -> strong
upvoted 2 times

🗨️ 👤 **SwePha** 1 year, 1 month ago

"Strong" is the answer as it needs guaranteed consistency
upvoted 3 times

HOTSPOT

-

You plan to create an Azure Cosmos DB for NoSQL container that will contain items in the following format.

```
{
  "userid": "user1@contoso.com",
  "id": "e379aea5-63f5-4623-92e9b-4cd9b33b91d5",
  "orderdate": "2022-11-01T02:00:00",
  "items": [
    {
      "sku": "11111",
      "qty": 1
    },
    {
      "sku": "11112",
      "qty": 5
    }
  ]
}
```

You need to define a query that will return the userid values of all the users who purchased an item that has a sku value of 11111.

How should you complete the query? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

SELECT c.userid

FROM c

WHERE EXISTS (

SELECT VALUE n

FROM

	▼
c.items	
c.items.sku	
n IN c.items	

WHERE

	▼
n	
n.items.sku	
n.sku	

= "11111"

}

Answer Area

Suggested Answer:



```
SELECT c.userid
FROM c
WHERE EXISTS(
    SELECT VALUE n
    FROM
        

|              |   |
|--------------|---|
|              | ▼ |
| c.items      |   |
| c.items.sku  |   |
| n IN c.items |   |


    WHERE
        

|             |   |
|-------------|---|
|             | ▼ |
| n           |   |
| n.items.sku |   |
| n.sku       |   |


        = "11111"
)
```

  **[Removed]** 9 months, 3 weeks ago

Provided answer is correct

upvoted 4 times

HOTSPOT

-

You have an Azure subscription that contains an Azure Cosmos DB for NoSQL account named account1 and a Log Analytics workspace named Workspace1. Workspace1 stores the logs of account1.

You need to identify which operations used the most request units per second (RU/s) during the last 24 hours.

How should you complete the query? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

▼

AzureActivity
AzureDiagnostics
AzureMetrics

```
| where ResourceProvider=="MICROSOFT.DOCUMENTDB" and Category==
```

```
| where TimeGenerated >= ago(1d)
```

▼

"ControlPlaneRequests"
"DataPlaneRequests"
"PartitionKeyStatistics"

```
| summarize max(responseLength_s), max(requestLength_s), max(requestCharge_s), count = count() by OperationName,  
requestResourceType_s, userAgent_s, collectionRid_s
```

Answer Area

▼

AzureActivity
AzureDiagnostics
AzureMetrics

Suggested Answer:


```
| where ResourceProvider=="MICROSOFT.DOCUMENTDB" and Category==
```

```
| where TimeGenerated >= ago(1d)
```

▼

"ControlPlaneRequests"
"DataPlaneRequests"
"PartitionKeyStatistics"

```
| summarize max(responseLength_s), max(requestLength_s), max(requestCharge_s), count = count() by OperationName,  
requestResourceType_s, userAgent_s, collectionRid_s
```

 **[Removed]** 8 months, 1 week ago

Provided answer is correct.

upvoted 3 times

You provision an Azure Cosmos DB for NoSQL container. You set the throughput to Autoscale, and the maximum request units per second (RU/s) to 20,000.

For how many RU/s will you be charged when the actual RU/s usage is zero?

- A. 0
- B. 200
- C. 2,000
- D. 4,000
- E. 10,000

Suggested Answer: A

Community vote distribution

C (100%)

🗨️ 👤 **Mangino** Highly Voted 1 year, 4 months ago

Selected Answer: C

The answer should be C - 2 000, this is because when we are using autoscale the container will not scale to 0, it will be at least 10% of your maximum throughput

<https://azure.microsoft.com/en-us/pricing/details/cosmos-db/autoscale-provisioned/>
upvoted 8 times

🗨️ 👤 **leoli9001** Highly Voted 1 year, 4 months ago

Correct Answer should be C. 2000, when autoscale is configured, MIN RU/s will be 10% of MAX RU/s
upvoted 5 times

🗨️ 👤 **Muteeb123** Most Recent 1 year ago

<https://learn.microsoft.com/en-us/azure/cosmos-db/autoscale-faq>
It should be charged $0.1 * \text{max RU/s}$
upvoted 1 times

You have an Azure Cosmos DB for NoSQL account that contains a database named DB1 and a container named Container1.

You need to manage the account by using the Azure Cosmos DB SDK.

What should you do?

- A. Create a container in DB1.
- B. List the physical partitions of Container1.
- C. Read a stored procedure in Container1.
- D. Create a user defined function (UDF) in Container1.

Suggested Answer: D

Community vote distribution

D (50%)

C (50%)

🗳️ 👤 **WimTS** 2 months, 2 weeks ago

Selected Answer: A

Creating a container is the only management action.

Stored procedures are only scoping 1 logical partion, an UDF can only be used in a query.

upvoted 1 times

🗳️ 👤 **YellowSky002** 5 months, 1 week ago

Selected Answer: A

Creating a container within a database is a clear example of managing the database using the SDK. It involves direct interaction with the database and influences its structure and organization.

upvoted 2 times

🗳️ 👤 **Examdumps2023** 1 year ago

Selected Answer: D

Answer: D

D. Create a user defined function (UDF) in Container1.

Here's why this is the most appropriate option:

Creating a UDF involves interacting with the Azure Cosmos DB SDK to manage the database.

UDFs are common tasks in database management, and they typically require SDK usage for implementation and deployment

upvoted 2 times

🗳️ 👤 **[Removed]** 1 year, 1 month ago

Selected Answer: C

To manage an Azure Cosmos DB account using the Azure Cosmos DB SDK, you would typically perform operations such as reading or writing data, or executing stored procedures. Therefore, the most appropriate action from the given options would be:

C. Read a stored procedure in Container1.

upvoted 3 times

HOTSPOT

-

You have an Azure Cosmos DB for NoSQL container that contains the following item.

```
{
  "id": "SalesOrder1",
  "PONumber": "P01",
  "OrderDateTime": "2022-11-21T00:00:00.0000000Z",
  "AccountNumber": "Acc1",
  "Items": [
    {
      "Id": 1,
      "OrderQty": 10,
      "ProductCode": "A111",
      "UnitPrice": 29.99
    }
  ]
}
```

You need to update the OrderQty value to 5 by using a patch operation.

How should you complete the JSON Patch document? To answer, select the appropriate options in the answer area.

Answer Area

```
[
  { "op": , "path": , "value": -5 }
]
```

Answer Area

Suggested Answer:

```
[
  { "op": , "path": , "value": -5 }
]
```

8fe085a 6 months, 3 weeks ago

The answer should be:

```
[{"op": "increment", "path": "/items/0/OrderQty", "value": -5}]
```

See: <https://learn.microsoft.com/en-us/azure/cosmos-db/partial-document-update>

Increment

This operator increments a field by the specified value. It can accept both positive and negative values. If the field doesn't exist, it creates the field and sets it to the specified value.

Add

If the target path specifies an element that already exists, its value is replaced.

The value in the question is -5 so using Add would set the node to -5 not 5.

upvoted 3 times

[Removed] 7 months, 3 weeks ago

Path option is wrong. Here is the correct JSON

```
[  
{  
  "op": "add",  
  "path": "/items/0/OrderQty",  
  "value": 5  
}  
]
```

upvoted 3 times

You plan to create an Azure Cosmos DB for NoSQL account that will have a single write region and three read regions.

You need to set the consistency level for the account. The solution must meet the following requirements:

- In the write region, writes must replicate synchronously across at least three replicas.
- In the read regions, reads must see writes in order for transactional batches.
- Throughput for reads and writes must be maximized.

Which consistency level should you select?

- A. Consistent Prefix
- B. Bounded Staleness
- C. Eventual
- D. Strong

Suggested Answer: A

Community vote distribution

D (100%)

🗨️ 👤 **WimTS** 2 months, 2 weeks ago

Selected Answer: A

Why not Consistent Prefix: reads must appear in order, and throughput must be maximized, which is not the case with Strong. In every region, writes are done to 3 replicas automatically.

upvoted 2 times

🗨️ 👤 **Examdumps2023** 1 year ago

Selected Answer: D

Strong:

Ensures that reads always return the most recent write.

Provides a linearizability guarantee, meaning writes are synchronously replicated to all replicas before acknowledgment.

Ensures that reads are guaranteed to see the most recent writes, which aligns with the requirement for transactional batches.

upvoted 1 times

🗨️ 👤 **[Removed]** 1 year, 1 month ago

Selected Answer: D

In the write region, writes must replicate synchronously across at least three replicas.

Only the Strong consistency provides the synch writing, all others are async. Therefore Strong is the answer.

upvoted 3 times

DRAG DROP -

You have an Azure Cosmos DB Core (SQL) API account that is configured for multi-region writes. The account contains a database that has two containers named container1 and container2.

The following is a sample of a document in container1:

```
{
  "customerId": 1234,
  "firstName": "John",
  "lastName": "Smith",
  "policyYear": 2021
}
```

The following is a sample of a document in container2:

```
{
  "gpsId": 1234,
  "latitude": 38.8951,
  "longitude": -77.0364
}
```

You need to configure conflict resolution to meet the following requirements:

- ⇒ For container1 you must resolve conflicts by using the highest value for policyYear.
- ⇒ For container2 you must resolve conflicts by accepting the distance closest to latitude: 40.730610 and longitude: -73.935242.
- ⇒ Administrative effort must be minimized to implement the solution.

What should you configure for each container? To answer, drag the appropriate configurations to the correct containers. Each configuration may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Select and Place:

Configurations

Last Write Wins (default) mode

Merge Procedures (custom) mode

An application that reads from the conflicts feed

Answer Area

Container1:

Container2:

Suggested Answer:**Configurations**

Last Write Wins (default) mode

Merge Procedures (custom) mode

An application that reads from the conflicts feed

Answer Area

Container1:

Container2:

Last Write Wins (default) mode

Merge Procedures (custom) mode

Box 1: Last Write Wins (LWW) (default) mode

Last Write Wins (LWW): This resolution policy, by default, uses a system-defined timestamp property. It's based on the time-synchronization clock protocol.

Box 2: Merge Procedures (custom) mode

Custom: This resolution policy is designed for application-defined semantics for reconciliation of conflicts. When you set this policy on your Azure Cosmos container, you also need to register a merge stored procedure. This procedure is automatically invoked when conflicts are detected under a database transaction on the server. The system provides exactly once guarantee for the execution of a merge procedure as part of the commitment protocol.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/conflict-resolution-policies> <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-manage-conflicts>

  **YJC**  1 year, 7 months ago

For the highest value wins, there is one description relate to LWW.

For API for NoSQL, this may also be set to a user-defined path with a numeric type. In a conflict, the highest value wins.

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/how-to-manage-conflicts?tabs=dotnetv2%2Capi-async%2Casync>
upvoted 8 times

  **azuredemo2022three**  1 year ago

Answer

Container1:

Last Write Wins (default) mode

Container2:



Merge Procedures (custom) mode

upvoted 7 times

  **Shaunp**  1 year, 8 months ago

what is the correct order Procedure and applicant?

upvoted 1 times

  **TimSss** 1 year, 9 months ago

it is explicitly stated that LWW used the DEFAULT (timestamp) mode, so increasing policyYear will not be used by LWW

upvoted 3 times

  **DrC** 1 year, 10 months ago

I don't see how the default (last one to write wins) would handle "the highest value for policyYear" if there was a conflict. Both should be procedures.

upvoted 3 times

  **ognamala** 1 year, 10 months ago

Both answers are correct, remember that a LWW policy can be configured to use any numeric path, not just the timestamp, so it can be configured with the /policyYear path where the highest number will win and it would satisfy the requirement

upvoted 5 times

  **Torent2005** 1 year, 9 months ago

Not really, default mode for LWW is based on timestamp.

"Last Write Wins (LWW): This resolution policy, by default, uses a system-defined timestamp property. It's based on the time-synchronization clock protocol. If you use the SQL API, you can specify any other custom numerical property (e.g., your own notion of a timestamp) to be used for conflict resolution. A custom numerical property is also referred to as the conflict resolution path."

<https://learn.microsoft.com/en-us/azure/cosmos-db/conflict-resolution-policies>

For Merge Procedures (custom) mode it's also not possible because you should set this policy during container creation. You can't do this for existing container.

"Custom conflict resolution policy is available only for SQL API accounts and can be set only at creation time. It is not possible to set a custom resolution policy on an existing container."

So the correct options for both containers is the last one - "an application that reads from the conflict feed."

upvoted 3 times

  **Torent2005** 1 year, 9 months ago

In general, you can't modify conflict policy after container creation.

upvoted 1 times

HOTSPOT -

You have an Azure Cosmos DB Core (SQL) API account named storage1 that uses provisioned throughput capacity mode. The storage1 account contains the databases shown in the following table.

Name	Throughput	Max request units per second (RU/s)	Geo-redundancy	Multi-region writes	Number of regions
db1	Autoscale	5,000	Disabled	Disabled	1
db2	Autoscale	8,000	Enabled	Enabled	3

The databases contain the containers shown in the following table.

Name	Database	Throughput
cn01	db1	Container - autoscale maximum RU/s of 10,000
cn02	db1	Database
cn03	db1	Database
cn04	db1	Database
cn05	db1	Database
cn11	db2	Database
cn12	db2	Database
cn13	db2	Database
cn14	db2	Database
cn15	db2	Database
cn16	db2	Database
cn17	db2	Database
cn18	db2	Database

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
At a minimum, you will be billed for 4,000 RU/s per hour for db1	<input type="radio"/>	<input type="radio"/>
The maximum throughput that can be consumed by cn11 is 400 RU/s	<input type="radio"/>	<input type="radio"/>
To db2, you can add a new container that uses database throughput	<input type="radio"/>	<input type="radio"/>

Suggested Answer:

Answer Area

Statements	Yes	No
At a minimum, you will be billed for 4,000 RU/s per hour for db1	<input type="radio"/>	<input checked="" type="radio"/>
The maximum throughput that can be consumed by cn11 is 400 RU/s	<input type="radio"/>	<input checked="" type="radio"/>
To db2, you can add a new container that uses database throughput	<input checked="" type="radio"/>	<input type="radio"/>

Box 1: No -

Four containers with 1000 RU/s each.

Box 2: No -

Max 8000 RU/s for db2. 8 containers, so 1000 RU/s for each container.

Box 3: Yes -

Max 8000 RU/s for db2. 8 containers, so 1000 RU/s for each container. Can very well add an additional container.



Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/plan-manage-costs> <https://azure.microsoft.com/en-us/pricing/details/cosmos-db/>

  **ognamala** Highly Voted 2 years, 10 months ago

Answers are correct but the reasoning for the first answer makes no sense - at minimum you will be charged $10\% \times 5000 = 500$ (db autoscale) + $10\% \times 10,000$ (container 1 autoscale) = 1500 in total

upvoted 21 times

  **Zaytsev** 2 years, 2 months ago

You are only counting one container. If you had the four left on db1, you get $4000 + 1500 = 5500$ minimum.

upvoted 2 times

  **Rifelo** 1 year ago

it's not correct i think.

the count is:

5000 ru for db1 and 5 container, so 1000 run x container.

We know that the minimun billed is $0.1 \times T_{max}$:

Each hour, you're billed for the highest throughput T the system scaled to within that hour. If your resource had no requests during the hour or didn't scale beyond $0.1 \times T_{max}$, you're billed for the minimum of $0.1 \times T_{max}$.

So the count is: $0.1 \times 1000 \text{ ru} \times 5 = 500$.

Autoscale is 10.000 ru so $0.1 \times 10.000 = 1000$.

total is 1500.

upvoted 2 times

  **biglebowski88** 6 months, 2 weeks ago

I believe the autoscaled 10,000 one takes precendent in billing over the shared ones so i think the min is just 10% of 10,000. 1000 RU/s

upvoted 1 times

  **p2006** Most Recent 2 weeks, 2 days ago

no, 500, here, talking about "db1" only, i thnk

no, 8000

yes

upvoted 1 times

  **AshishKu** 1 year, 8 months ago

For db1 minimum RUs 1500 will be billed. 500 RU at database level and 1000 at container level.

upvoted 1 times

HOTSPOT -

You have a database named telemetry in an Azure Cosmos DB Core (SQL) API account that stores IoT data. The database contains two containers named readings and devices.

Documents in readings have the following structure.

- ⇒ id
- ⇒ deviceid
- ⇒ timestamp
- ⇒ ownerid
- ⇒ measures (array)
- type
- value
- metricid

Documents in devices have the following structure.

- ⇒ id
- ⇒ deviceid
- ⇒ owner
- ownerid
- emailaddress
- name
- ⇒ brand
- ⇒ model

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
To return for all devices owned by a specific emailaddress, multiple queries must be performed	<input type="radio"/>	<input type="radio"/>
To return deviceid, ownerid, timestamp, and value for a specific metricid, a join must be performed	<input type="radio"/>	<input type="radio"/>
To return deviceid, ownerid, emailaddress, and model, a join must be performed	<input type="radio"/>	<input type="radio"/>

Suggested Answer:

Answer Area

Statements	Yes	No
To return for all devices owned by a specific emailaddress, multiple queries must be performed	<input checked="" type="radio"/>	<input type="radio"/>
To return deviceid, ownerid, timestamp, and value for a specific metricid, a join must be performed	<input type="radio"/>	<input checked="" type="radio"/>
To return deviceid, ownerid, emailaddress, and model, a join must be performed	<input type="radio"/>	<input checked="" type="radio"/>

Box 1: Yes -

Need to join readings and devices.

Box 2: No -

Only readings is required. All required fields are in readings.

Box 3: No -

Only devices is required. All required fields are in devices.

  **jdsherrington** Highly Voted 1 year, 10 months ago

Answer is NYN.

1. N: The readings container does not need to be read to return all deviceids associated with a particular emailaddress, so only one query is needed:

```
SELECT *
```

```
FROM devices d
```

```
WHERE d.owner.emailaddress = _____
```

2. Y: measures is an array, and to filter or project fields from an array, you need to use join to create a cross join with the array.

```
SELECT r.deviceid, r.ownerid, r.timestamp, m.value
```

```
FROM readings r
```

```
JOIN m IN r.measures
```

```
WHERE m.metricid = _____
```

3. N: owner is an object, not an array, so you can directly access the properties. deviceid, ownerid, emailaddress and model all exist in the devices container, so the readings container is not needed.

```
SELECT d.deviceid, d.owner.ownerid, d.owner.emailaddress, d.model
```



```
FROM devices d
```

upvoted 25 times

  **WimTS** 2 months, 2 weeks ago

exact, only it should be d.name.model

upvoted 1 times

  **SwePha** 1 year, 7 months ago

Awesome explanation

upvoted 2 times

  **grada** Highly Voted 2 years, 11 months ago

Who ever wrote the suggested responses has no idea of what a JOIN does in a document database... it has nothing to do with joining documents, but is an inner-document join.

It's NYN if the first questions means returning only devices, and none of the readings. It's YYN otherwise, because readings are in a separate container, and should be fetched using a second query.

upvoted 17 times

  **Shiven** Most Recent 1 year, 10 months ago

N - Device container has all required info.

N - In Reading container we have everything.

Y - Need to join 2 containers.

upvoted 4 times

  **azuredemo2022three** 2 years ago

Answer YYN

upvoted 1 times

  **avocacao** 2 years, 6 months ago

Y - requires 2 requests to get from 2 containers

Y - need join to get 'value' in measures array

Y - need join to get 'ownerid' and 'emailaddress' in owner array

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/query/join>

upvoted 4 times

  **Alex22022** 2 years, 5 months ago

Third - No; Devices container: select c.deviceid, c.owner.ownerid, c.owner.emailaddress, c.model from c

upvoted 1 times

  **Internal_Koala** 2 years, 9 months ago

No. Yes. Yes.

=====

First: No

```
SELECT *  
FROM d  
WHERE d.emailaddress =
```

"To return (?) for all devices ...". Sounds like the word "metrics" might be missing. If so, the answer is Yes and a second query must be added.

=====

```
SELECT d.deviceid  
FROM d  
WHERE d.emailaddress = ?
```

```
SELECT *  
FROM m  
WHERE m.deviceid = @deviceid
```

=====

Second: Yes

```
SELECT m.deviceid  
FROM m  
WHERE m.metricid = ?
```

```
SELECT d.deviceid, o.ownerid, d.timestamp  
FROM d  
JOIN o IN d.owner  
WHERE d.deviceid = @deviceid
```

=====



Third: Yes

```
SELECT d.deviceid, o.ownerid, d.emailaddress, d.model  
FROM d  
JOIN o IN d.owner  
upvoted 9 times
```

  **susejzpol** 2 years, 7 months ago

i think like you.

upvoted 1 times

  **nope1234567** 2 years, 6 months ago

I disagree on the second one. All fields to return are in the Readings container, why would you need to join the second container ? FK are sufficients here

upvoted 2 times

  **kdsingh** 3 years, 1 month ago

The correct answer is YYN

- Yes: you need 2 requests, because the data is in 2 different containers.

- Yes: you need join to get the data from Array

- No: No joins required

upvoted 7 times

  **ognamala** 2 years, 10 months ago

Why would the second one be yes ? We dont need any data from the array.

upvoted 3 times

  **Alex22022** 2 years, 5 months ago

Because you need value and metricid properties from the measures array.

upvoted 1 times

The settings for a container in an Azure Cosmos DB Core (SQL) API account are configured as shown in the following exhibit.

Settings

Indexing Policy

Time to Live

- ☐ Off
- ☒ On (no default)
- ☐ On

Geospatial Configuration

- ☒ Geography
- ☐ Geometry

Partition key

/productName

Which statement describes the configuration of the container?

- A. All items will be deleted after one year.
- B. Items stored in the collection will be retained always, regardless of the items time to live value.
- C. Items stored in the collection will expire only if the item has a time to live value.
- D. All items will be deleted after one hour.

Suggested Answer: C

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-live>

 **azuredemo2022three** Highly Voted 1 year ago

C. Items stored in the collection will expire only if the item has a time to live value.
upvoted 9 times

 **Tuopikson** Most Recent 3 months, 3 weeks ago

Selected Answer: C

Item stored in the collection will expire only if the item has a time to live value.

upvoted 1 times

You have an Azure Cosmos DB Core (SQL) API account named account1 that is configured for automatic failover. The account1 account has a single read-write region in West US and a read region in East US.

You run the following PowerShell command.

```
Update-AzCosmosDBAccountFailoverPriority -ResourceGroupName `rg1` -Name `account1` -FailoverPolicy @(`East US`, `West US`)
```

What is the effect of running the command?

- A. The account will be unavailable to writes during the change
- B. The provisioned throughput for account1 will increase
- C. The account will be configured for multi-region writes
- D. A manual failover will occur

Suggested Answer: A

Note: The Update-AzCosmosDBAccountFailoverPriority command updates Failover Region Priority of a CosmosDB Account.

Parameter -FailoverPolicy is an array of strings having region names, ordered by failover priority. E.g eastus, westus

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/high-availability>

Community vote distribution



christosst 2 years, 6 months ago

I think is D - West US is the failoverPriority=0 right?

Any change to a region with failoverPriority=0 triggers a manual failover and can only be done to an account configured for manual failover. Changes to all other regions simply changes the failover priority for an Azure Cosmos DB account.

<https://learn.microsoft.com/en-us/azure/cosmos-db/scripts/powershell/common/failover-priority-update>
upvoted 19 times

harass 1 year, 9 months ago

This answer is incorrect because changing all regions will only change the priority.

<https://learn.microsoft.com/en-us/azure/cosmos-db/scripts/powershell/common/failover-priority-update>
(Changes to all other regions simply changes the failover priority for an Azure Cosmos DB account.)

No failover will occur because the powershell command change will trigger all regions.

upvoted 2 times

IDATA 2 years, 9 months ago

I think its wrong, the answer is D

Because just have 2 failover regions, if u flip this 2 this gona make a Manual Failover

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/manage-with-powershell>
upvoted 8 times

codingdown 2 years, 7 months ago

you are not performing the failover, just deciding the autofailovr policy

upvoted 1 times

Alex22022 2 years, 4 months ago

According to the link below, a manual failover is performed if you change one of the regions to priority = 0.



<https://learn.microsoft.com/en-us/training/modules/write-scripts-for-azure-cosmos-db-sql-api/7-initiate-failovers>
upvoted 2 times

monniq 1 year, 9 months ago

The accounts are configured for automatic failover. In order for manual failover to happen, the account with priority 0 must be configured for manual failover.

"Any change to a region with failoverPriority=0 triggers a manual failover and can only be done to an account configured for manual failover."

<https://learn.microsoft.com/en-us/azure/cosmos-db/scripts/powershell/common/failover-priority-update>
upvoted 1 times

  **harass** 1 year, 9 months ago

This answer is incorrect because changing all regions will only change the priority.

<https://learn.microsoft.com/en-us/azure/cosmos-db/scripts/powershell/common/failover-priority-update>

(Changes to all other regions simply changes the failover priority for an Azure Cosmos DB account.)

upvoted 1 times

  **Tuopikson** Most Recent 4 months, 1 week ago



Selected Answer: A

When you run the command `Update-AzCosmosDBAccountFailoverPriority` with the specified failover policy, you are changing the priority order of the regions. In this case, you are setting the East US region as the primary (read-write) region and the West US region as the secondary (read-only) region.

The effect of this command is: A. The account will be unavailable to writes during the change.

Changing the failover priority order involves a brief period during which write operations are unavailable. This is because the primary region is being switched, and during the transition, the account cannot accept write operations until the new primary region is fully operational.

upvoted 1 times

  **szkielet** 4 months, 2 weeks ago

Selected Answer: D

Imagine your Azure Cosmos DB account as a bustling train station. Right now, West US is the central hub where all the trains (writes) depart from, and East US is a satellite station handling passengers (reads). By running this PowerShell command:

powershell

`Update-AzCosmosDBAccountFailoverPriority -ResourceGroupName `rg1` -Name `account1` -FailoverPolicy @('East US', `West US`)`

You're essentially rerouting the main line to East US, making it the new central hub. Here's what happens:

Effect of the Command:

Manual Failover Triggered: You're updating the failover priority, placing East US above West US. Since your account is configured for automatic failover, this change initiates a manual failover to East US. The write region shifts from West US to East US.

Answer: D. A manual failover will occur

upvoted 1 times

  **poesklap** 1 year ago

Selected Answer: D

A. The account will be unavailable to writes during the change:

This is partially true. During the change, there might be a brief period where writes are unavailable as the system transitions the write region. However, this downtime is usually minimal.

B. The provisioned throughput for account1 will increase:

No, changing the failover priority does not affect the provisioned throughput.

C. The account will be configured for multi-region writes:

No, the command does not configure the account for multi-region writes. Multi-region writes need to be explicitly configured.

D. A manual failover will occur:

Yes, the command triggers a failover process by changing the priority, effectively causing a manual failover to the new region.

upvoted 1 times

  **Examdumps2023** 1 year ago

Selected Answer: D

By changing the failover priority, you are effectively initiating a manual failover to the new primary region.



upvoted 1 times

  **Blubb1860** 1 year, 6 months ago

Selected Answer: D

D is correct, setting failover priority 0 means it's the new write region

upvoted 2 times

  **Tito28** 1 year, 9 months ago

It can't be D cause the account is configured for automatic failover.

<https://learn.microsoft.com/en-us/azure/cosmos-db/scripts/powershell/common/failover-priority-update>



The right option is C.

The command Update-AzCosmosDBAccountFailoverPriority with the specified failover policy updates the failover priority of the Cosmos DB account.

In this case, the failover policy includes both 'East US' and 'West US' regions, indicating that the account will be configured for multi-region writes.

The command does not make the account unavailable for writes during the change, increase the provisioned throughput, or trigger a manual failover.

upvoted 1 times



  **harass** 1 year, 9 months ago

The failover occurs, but D's answer says that a manual failover occurs. In this case, the failover that occurred was a service-managed failover, not a manual failover, so D is not an answer.

I also did not find the same results with answers B and C.

The correct answer should be A.

upvoted 1 times

  **harass** 1 year, 9 months ago

This answer reflects the results of testing after configuring the environment.



upvoted 1 times

  **Feanorich** 1 year, 10 months ago

Selected Answer: D

It is D

upvoted 2 times



  **harass** 1 year, 9 months ago

This answer is incorrect because changing all regions will only change the priority.

<https://learn.microsoft.com/en-us/azure/cosmos-db/scripts/powershell/common/failover-priority-update>

(Changes to all other regions simply changes the failover priority for an Azure Cosmos DB account.)

upvoted 1 times

  **azuredemo2022three** 2 years ago

Selected Answer: C

Selected Answer: C

upvoted 2 times

  **azuredemo2022three** 2 years ago

Selected Answer: C



upvoted 1 times

  **imando** 2 years, 1 month ago

Selected Answer: A

Its A as an answer

upvoted 3 times

  **arnabdt** 2 years, 3 months ago

Selected Answer: C

C. The account will be configured for multi-region writes.

The Update-AzCosmosDBAccountFailoverPriority command with the -FailoverPolicy parameter updates the failover policy for the Cosmos DB account by specifying the order in which the regions are selected for failover.

In this case, the command updates the failover policy for account1 to include both West US and East US regions, in that order. This means that if the primary (read-write) region in West US becomes unavailable, Cosmos DB will automatically failover to the secondary (read-only) region in East US.

The command does not affect the availability of the account or the provisioned throughput. It only updates the failover policy to allow for multi-region writes in case of a failover.

Option C is correct because the command enables multi-region writes in the failover policy. Options A and B are incorrect because the command does not affect the account's availability or provisioned throughput. Option D is incorrect because the command does not trigger a manual failover.

upvoted 3 times

🗨️ 👤 **Ranzzzan** 2 years, 4 months ago

IMO its A , The Azure Cosmos DB account must be configured for manual failover for manual failover to succeed

upvoted 2 times

🗨️ 👤 **virgilpza** 2 years, 4 months ago

according to <https://learn.microsoft.com/en-us/azure/cosmos-db/scripts/powershell/common/failover-priority-update>, thats what i believe to.

"Any change to a region with failoverPriority=0 triggers a manual failover and can only be done to an account configured for manual failover.

Changes to all other regions simply changes the failover priority for an Azure Cosmos DB account."

upvoted 1 times

🗨️ 👤 **Internal_Koala** 2 years, 9 months ago

As per exact example on the source:

<https://learn.microsoft.com/en-us/azure/cosmos-db/sql/manage-with-powershell#trigger-manual-failover>

From this website and according to A:

"If you perform a manual failover operation while an asynchronous throughput scaling operation is in progress, the throughput scale-up operation will be paused. It will resume automatically when the failover operation is complete."

1. Has to be an asynchronous throughput scaling operation

2. It will continue after failover completes.

upvoted 1 times

🗨️ 👤 **ExamsBertia** 2 years, 9 months ago

They said: "a single read-write region in West US and a read region in East US"

If you flip the FailoverPolicy to "East US" you can't write because it's a read region. So i think A is a correct answer.

upvoted 1 times

HOTSPOT -

You are developing an application that will connect to an Azure Cosmos DB Core (SQL) API account. The account has a single read-write region and one additional read region. The regions are configured for automatic failover.

The account has the following connection strings. (Line numbers are included for reference only.)

```

01 {
02   "connectionStrings": [
03     {
04       "connectionString":
05       "AccountEndpoint=https://constosodbaccount.documents.azure.com:443/;
06       AccountKey=MwUgRnGti4vErT2rfPPFdTFFyI9KyI9Kbe1RPGv7OQdHo6VZ2i45TcJzrd4J80zYxrEATzyZh0m1nJaNFA==;",
07       "description": "Primary SQL Connection String"
08     },
09     {
10       "connectionString":
11       "AccountEndpoint=https://constosodbaccount.documents.azure.com:443/;
12       AccountKey=gfThRnGti4vErT2rfPPFdTFFyI43529Kbe1RPGv7OQdHo6VZ2i45TcJzrd4J80zYxrfatzyZh0m1nJaNFA==;",
13       "description": "Secondary SQL Connection String"
14     },
15     {
16       "connectionString":
17       "AccountEndpoint=https://constosodbaccount.documents.azure.com:443/;
18       AccountKey=WGykBc1PHJoos6MdErT2rfPPFx9yI9Kbe1RPGv7OQlIQwQNxq6QdOXjxgyLLebXBp8uJu7FyJy3Uv1vuK2A==;",
19       "description": "Primary Read-Only SQL Connection String"
20     },
21     {
22       "connectionString":
23       "AccountEndpoint=https://constosodbaccount.documents.azure.com:443/;
24       AccountKey=k2DZIOoY4Jc7QeUJqVGH3csda6EyI9Kbe1RPGv7OQErT2rfPPFtbwTPfKAg19zVxCOMDNn8xPpQrednVYcQ==;",
25       "description": "Secondary Read-Only SQL Connection String"
26     }
27   ]
28 }

```

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 the primary write region fails, applications that write to the database must use a different connection string to continue to use the service	<input type="radio"/>	<input type="radio"/>
The Primary Read-Only SQL Connection String and the Secondary Read-Only SQL Connection String will connect to different regions from an application running in the East US Azure region	<input type="radio"/>	<input type="radio"/>
Applications can choose from which region they read by setting the PreferredLocations property within their connection properties	<input type="radio"/>	<input type="radio"/>

Suggested Answer:

Answer Area

Statements	Yes	No
If the primary write region fails, applications that write to the database must use a different connection string to continue to use the service	<input type="radio"/>	<input checked="" type="radio"/>
The Primary Read-Only SQL Connection String and the Secondary Read-Only SQL Connection String will connect to different regions from an application running in the East US Azure region	<input type="radio"/>	<input checked="" type="radio"/>
Applications can choose from which region they read by setting the PreferredLocations property within their connection properties	<input checked="" type="radio"/>	<input type="radio"/>

Box 1: No -

The same connection string can still be used.

Azure Cosmos DB Core read write region connection strings

Azure Cosmos DB "primary write region" failover "connection string"

Box 2: No -

The AccountEndpoint for both are the same.



Box 3: Yes -

The ConnectionPolicy.PreferredLocations property gets and sets the preferred locations (regions) for geo-replicated database accounts in the Azure Cosmos DB service. For example, "East US" as the preferred location.

When EnableEndpointDiscovery is true and the value of this property is non-empty, the SDK uses the locations in the collection in the order they are specified to perform operations, otherwise if the value of this property is not specified, the SDK uses the write region as the preferred location for all operations.

Reference:

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.documents.client.connectionpolicy.preferredlocations>

  **dtodorov** 9 months, 2 weeks ago

Yes, I think answer is correct - No, No, Yes

upvoted 3 times

You have a global ecommerce application that stores data in an Azure Cosmos DB Core (SQL) API account. The account is configured for multi-region writes.

You need to create a stored procedure for a custom conflict resolution policy for a new container. In the event of a conflict caused by a deletion, the deletion must always take priority.

Which parameter should you check in the stored procedure function?

- A. isTombstone
- B. conflictingItems
- C. existingItem
- D. incomingItem

Suggested Answer: A

isTombstone: Boolean indicating if the incomingItem is conflicting with a previously deleted item. When true, existingItem is also null.

Incorrect:

conflictingItems: Array of the committed version of all items in the container that are conflicting with incomingItem on ID or any other unique index properties. existingItem: The currently committed item. This value is non-null in an update and null for an insert or deletes. incomingItem: The item being inserted or updated in the commit that is generating the conflicts. Is null for delete operations.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-manage-conflicts>

Community vote distribution


A (100%)

 **azuredemo2022three** Highly Voted 1 year, 6 months ago

A. isTombstone
upvoted 5 times

 **Blubb1860** Most Recent 1 year ago

Selected Answer: A
A isTombstone
upvoted 3 times

 **virgilpza** 1 year, 10 months ago

A is the correct answer.
See <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/how-to-manage-conflicts?tabs=dotnetv2%2Capi-async%2Casync>
upvoted 2 times

You have an Azure Cosmos DB Core (SQL) API account named account1 that supports an application named App1. App1 uses the consistent prefix consistency level.

You configure account1 to use a dedicated gateway and integrated cache.

You need to ensure that App1 can use the integrated cache.

Which two actions should you perform for App1? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Change the consistency level of requests to session.
- B. Change the account endpoint to `http://account1.documents.azure.com`.
- C. Change the account endpoint to `http://account1.sqlx.cosmos.azure.com`.
- D. Change the connection mode to direct.
- E. Change the consistency level of requests to strong.

Suggested Answer: AC

C: Modify your application's connection string to use the new dedicated gateway endpoint.

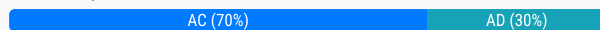
All dedicated gateway connection strings follow the same pattern. Remove `documents.azure.com` from your original connection string and replace it with `sqlx.cosmos.azure.com`. A dedicated gateway will always have the same connection string, even if you remove and reprovision it.

A: You must adjust the request consistency to session or eventual. If not, the request will always bypass the integrated cache.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-configure-integrated-cache>

Community vote distribution



TRUESON Highly Voted 1 year, 8 months ago

All dedicated gateway connection strings follow the same pattern. Remove `documents.azure.com` from your original connection string and replace it with `sqlx.cosmos.azure.com`

A

C

upvoted 13 times

Tuopikson Most Recent 3 months, 3 weeks ago

Selected Answer: AC

The integrated cache in Azure Cosmos DB can be used only when the consistency level is set to either session or consistent prefix. Since App1 initially uses the consistent prefix consistency level, you need to adjust it to session (A), as session consistency is supported by the integrated cache.

The dedicated gateway and its integrated cache require using a specific endpoint format (`sqlx.cosmos.azure.com`) to access the cache. So, you need to change the account endpoint to `http://account1.sqlx.cosmos.azure.com` (C).

Why not the other options?

B. Change the account endpoint to `http://account1.documents.azure.com`: This is the default endpoint format for Cosmos DB but does not support the integrated cache.

D. Change the connection mode to direct: The connection mode must remain as gateway when using the integrated cache.

E. Change the consistency level of requests to strong: The integrated cache does not support strong consistency.

upvoted 1 times

YellowSky002 4 months, 4 weeks ago

Selected Answer: AB

I change my answer.

The integrated cache uses the dedicated gateway within your Azure Cosmos DB account.

The integrated cache only works with the Direct connection mode in the Cosmos DB SDK. Make sure App1 is configured to use this mode. While Consistent Prefix works with integrated cache, yet Microsoft recommends using Session consistency for optimal cache hit rates. If your application's design allows, consider switching to Session consistency to potentially further improve performance.

upvoted 1 times



  **YellowSky002** 5 months, 3 weeks ago

Selected Answer: AD

The integrated cache only works with the Direct connection mode in the Cosmos DB SDK. Make sure App1 is configured to use this mode.



While Consistent Prefix works with integrated cache, yet Microsoft recommends using Session consistency for optimal cache hit rates. If your application's design allows, consider switching to Session consistency to potentially further improve performance. <https://learn.microsoft.com/en-us/azure/cosmos-db/integrated-cache>

upvoted 1 times

  **RosaHutor** 3 months, 4 weeks ago

Thanks AI, but you are totally wrong. To use integrated cache you definitely need to use Gateway mode instead of Direct

upvoted 1 times

  **3709334** 6 months, 1 week ago

Selected Answer: AC

AC are the correct options



upvoted 1 times

  **Blubb1860** 1 year ago

Selected Answer: AC

A and C is correct. The endpoint needs to change and only Session and eventual consistency levels are supported

upvoted 3 times

  **monniq** 1 year, 3 months ago

It's A and C. D is not an answer because if you choose to connect with direct mode your requests will not use the dedicated gateway or the integrated cache.



<https://learn.microsoft.com/en-us/azure/cosmos-db/dedicated-gateway#connect-to-azure-cosmos-db-using-direct-mode>

upvoted 4 times

  **Emil_Topics** 1 year, 4 months ago

The answer is correct A and C

upvoted 4 times

  **[Removed]** 1 year, 4 months ago

Selected Answer: AC

*All dedicated gateway connection strings follow the same pattern. Remove documents.azure.com from your original connection string and replace it with sqlx.cosmos.azure.com"

<https://learn.microsoft.com/en-us/azure/cosmos-db/how-to-configure-integrated-cache?tabs=dotnet#configuring-the-integrated-cache>

*"You must ensure the request consistency is session or eventual"

<https://learn.microsoft.com/en-us/azure/cosmos-db/how-to-configure-integrated-cache?tabs=dotnet#adjust-request-consistency>

*Connection for dedicated gateway is "gateway" mode

<https://learn.microsoft.com/en-us/azure/cosmos-db/dedicated-gateway#connect-to-azure-cosmos-db-using-gateway-mode>

<https://learn.microsoft.com/en-us/azure/cosmos-db/how-to-configure-integrated-cache?tabs=dotnet#verify-cache-hits>

upvoted 4 times

  **azuredemo2022three** 1 year, 6 months ago

Selected Answer: AD

Selected Answer: AD

upvoted 1 times

  **azuredemo2022three** 1 year, 6 months ago

Selected Answer: AD

upvoted 1 times

  **[Removed]** 1 year, 8 months ago



Selected Answer: AD

To use the integrated cache with an Azure Cosmos DB Core (SQL) API account, you need to use the "Gateway" connection mode and the "Session" or "Consistent Prefix" consistency levels. Therefore, you should perform the following actions for App1:

A. Change the consistency level of requests to session.

D. Change the connection mode.

upvoted 1 times

  **arnabdt** 1 year, 9 months ago

Selected Answer: AD

To ensure that App1 can use the integrated cache, you need to configure the Cosmos DB client SDK to use the Direct connection mode and set the consistency level to session. This is because the integrated cache is available only in the Direct connection mode and the session consistency level is recommended to maximize cache hit rates.

upvoted 1 times

You have an Azure Cosmos DB Core (SQL) API account named account1 that has a single read-write region and one additional read region. Account1 uses the strong default consistency level. You have an application that uses the eventual consistency level when submitting requests to account1. How will writes from the application be handled?

- A. Writes will use the eventual consistency level.
- B. Azure Cosmos DB will reject writes from the application.
- C. Writes will use the strong consistency level.
- D. The write order is not guaranteed during replication.

Suggested Answer: C

Overriding the default consistency level only applies to reads within the SDK client. An account configured for strong consistency by default will still write and replicate data synchronously to every region in the account. When the SDK client instance or request overrides this with Session or weaker consistency, reads will be performed using a single replica.

Reference:

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

Community vote distribution

C (100%)

 **TRUESON** Highly Voted 1 year, 8 months ago
C

Overriding the default consistency level only applies to reads within the SDK client. An account configured for strong consistency by default will still write and replicate data synchronously to every region in the account. <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/how-to-manage-consistency?tabs=portal%2Cdotnetv2%2Capi-async#override-the-default-consistency-level>
upvoted 7 times

 **Blubb1860** Most Recent 1 year ago

Selected Answer: C

C, Overriding the default consistency level only applies to reads
upvoted 2 times

 **azuredemo2022three** 1 year, 6 months ago

C. Writes will use the strong consistency level.
upvoted 4 times

HOTSPOT

You have a multi-region Azure Cosmos DB account named account1 that has a default consistency level of strong.

You have an app named App1 that is configured to request a consistency level of session.

How will the read and write operations of App1 be handled? To answer, select the appropriate options in the answer area.

Answer Area

Write operations will:

- ☐ Be applied only to the replica to which App1 is connected
- ☐ Write and replicate data to every region asynchronously
- ☐ Write and replicate data to every region synchronously

Read operations will be performed by the replica:

- ☐ To which App1 is connected
- ☐ That has the lowest estimated latency to the client
- ☐ That responds first

Answer Area

Suggested Answer:

Write operations will:

- ☐ Be applied only to the replica to which App1 is connected
- ☐ Write and replicate data to every region asynchronously
- ☒ Write and replicate data to every region synchronously

Read operations will be performed by the replica:

- ☒ To which App1 is connected
- ☐ That has the lowest estimated latency to the client
- ☐ That responds first

 **azuredemo2022three** Highly Voted 1 year, 6 months ago

Answers

Write and replicate data to every region synchronously.

To which App1 is connected.

upvoted 11 times

 **azuredemo2022three** 1 year, 5 months ago

Be applied only to the replica to which App1 is connected

and

To which App1 is connected

upvoted 1 times

 **Blubb1860** 1 year ago

If it will only write to this one replica, that would mean, there is no consistency at all...

It's writing synchronously because the strong consistency mode can't be overridden for writes

upvoted 1 times

 **meiwein** Highly Voted 1 year, 5 months ago

"An account configured for strong consistency by default will still write and replicate data synchronously to every region in the account. When the SDK client instance or request overrides this with Session or weaker consistency, reads will be performed using a single replica."

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

upvoted 5 times

You plan to create an Azure Cosmos DB account that will use the NoSQL API.

You need to create a grouping strategy for items that will be stored in the account. The solution must ensure that write and read operations on the items can be performed within the same transaction.

What should you use to group the items?

- A. logical partitions
- B. physical partitions
- C. databases
- D. containers

Suggested Answer: A

Community vote distribution

A (100%)

🗳️ 👤 **Blubb1860** 1 year ago

Selected Answer: A

A, transactional batches only work within the same logical partition
upvoted 2 times

🗳️ 👤 **SwePha** 1 year, 1 month ago

Selected Answer: A

Azure Cosmos DB supports full ACID compliant transactions with snapshot isolation for operations within the same logical partition key.

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/how-to-manage-consistency?tabs=portal%2Cdotnetv2%2Capi-async#override-the-default-consistency-level>
upvoted 2 times

🗳️ 👤 **dksk** 1 year, 1 month ago

Selected Answer: A

A selected
upvoted 2 times

You have an Azure Cosmos DB database that contains a container named container1. The container1 container is configured with a maximum of 20,000 RU/s and currently contains 240 GB of data.

You need to estimate the costs of container1 based on the current usage.

How many RU/s will be charged?

- A. 240
- B. 4,000
- C. 20,000
- D. 24,000

Suggested Answer: B

Community vote distribution

C (100%)

🗳️ 👤 **3709334** 6 months ago

Selected Answer: A

1GB = 1 RU. Based on current usage it should be 240 RUs

upvoted 1 times

🗳️ 👤 **RosaHutor** 3 months, 4 weeks ago

Wrong. Storage (GB) is billed separately and not related to RUs at all. The answer is C

upvoted 2 times

🗳️ 👤 **[Removed]** 10 months, 1 week ago

Selected Answer: C

C selected

upvoted 3 times

🗳️ 👤 **Blubb1860** 1 year ago

Selected Answer: C

C, if 20 000 RU/s are fixed (no autoscaling), then its 20 000 RU/s. Storage would need to be higher than 20 000 GB to beat the 20 000 RU/s.

upvoted 3 times

🗳️ 👤 **RosaHutor** 3 months, 4 weeks ago

Answer is correct, but you understanding is wrong. Storage (GB) is billed separately and not related to RUs at all

upvoted 1 times

🗳️ 👤 **dkssk** 1 year, 1 month ago

Selected Answer: C

C selected

upvoted 3 times

HOTSPOT

-

You have an Azure Cosmos DB for NoSQL account named account1 in a resource group named RG1. The Azure regions for account1 are shown in the following table.

Name	Type
East US (eastus)	Write
Central US (centralus)	Read
West US (westus)	Read

You need to fail over account1 from the East US region to the Central US region by using Azure Command-Line Interface (CLI).

How should you complete the command? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
az cosmosdb 

▼



failover-priority-change  
locations  
network-rule  
update

 --name account1 --resource-group rg1  


▼



--enable-automatic-failover  
--failover-policies  
failoverPriority=0

 "centralus=0" "westus=1" "eastus=2"
```

Answer Area

Suggested Answer:

```
az cosmosdb 

▼



failover-priority-change  
locations  
network-rule  
update

 --name account1 --resource-group rg1  


▼



--enable-automatic-failover  
--failover-policies  
failoverPriority=0

 "centralus=0" "westus=1" "eastus=2"
```

 [Removed] 8 months, 1 week ago

Answer is correct.

upvoted 1 times

You have an Azure Cosmos DB for NoSQL account that uses the default consistency level.

How can the consistency level be modified as part of a query request?

- A. stronger consistency for write operations
- B. weaker consistency for write operations
- C. stronger consistency for read operations
- D. weaker consistency for read operations

Suggested Answer: D

Community vote distribution

C (100%)

🗳️ 👤 **p2006** 3 weeks, 2 days ago

Selected Answer: D

weaker, read

<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/how-to-manage-consistency?tabs=portal%2Cdotnetv2%2Capi-async#override-the-default-consistency-level>

> Consistency can only be relaxed at the SDK instance or request level.

> Overriding the default consistency level only applies to reads within the SDK client.

upvoted 1 times

🗳️ 👤 **WimTS** 2 months, 2 weeks ago

Selected Answer: D

Write cannot be overridden

Read can only be weaker

upvoted 1 times

🗳️ 👤 **3709334** 6 months ago

Selected Answer: D

as part of RequestOptions can only override to weaker consistency. For overriding to stronger consistency need to change at account level through cosmos sdk client

upvoted 1 times

🗳️ 👤 **biglebowski88** 6 months, 2 weeks ago

Selected Answer: D

You can only override to weaker consistency in the request. So D

upvoted 2 times

🗳️ 👤 **GermanGerman** 10 months, 1 week ago

Selected Answer: C

Answer C

upvoted 1 times

You have an Azure Cosmos DB Core (SQL) API account that uses a custom conflict resolution policy. The account has a registered merge procedure that throws a runtime exception. The runtime exception prevents conflicts from being resolved. You need to use an Azure function to resolve the conflicts. What should you use?

- A. a function that pulls items from the conflicts feed and is triggered by a timer trigger
- B. a function that receives items pushed from the change feed and is triggered by an Azure Cosmos DB trigger
- C. a function that pulls items from the change feed and is triggered by a timer trigger
- D. a function that receives items pushed from the conflicts feed and is triggered by an Azure Cosmos DB trigger

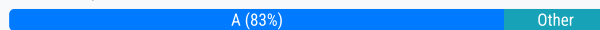
Suggested Answer: D

The Azure Cosmos DB Trigger uses the Azure Cosmos DB Change Feed to listen for inserts and updates across partitions. The change feed publishes inserts and updates, not deletions.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-cosmosdb>

Community vote distribution



ognamala Highly Voted 2 years, 4 months ago

Selected Answer: A

Correct answer should be "A" - we need to resolve conflicts, hence we definitely need to read the conflicts feed, hence answers B and C are immediately eliminated as they are pull from the changes feed, answer D mentions an Azure Cosmos DB trigger, but this is only for the change feed, not for the conflicts feed, hence A is the correct answer since there is no trigger mechanism for the conflict feed in Cosmos DB
upvoted 16 times

Blubb1860 Most Recent 1 year ago

Selected Answer: D

Conflict feed and CosmosDB trigger
upvoted 1 times

Bviljoen 1 year, 3 months ago

Answer is A:

<https://learn.microsoft.com/en-us/azure/cosmos-db/conflict-resolution-policies#conflict-resolution-policies>

If you configure your container with the custom resolution option, and you fail to register a merge procedure on the container or the merge procedure throws an exception at runtime, the conflicts are written to the conflicts feed. Your application then needs to manually resolve the conflicts in the conflicts feed. To learn more, see examples of how to use the custom resolution policy and how to use the conflicts feed.
upvoted 1 times

virgilpza 1 year, 10 months ago

Correct answer is A
upvoted 1 times

klepper 2 years, 3 months ago

Selected Answer: A

Correct answer is A
upvoted 1 times

Shiggi 2 years, 3 months ago

Selected Answer: A

Correct answer is A: all conflicts related data is stored in the conflicts feed, you have to orchestrate the function using a timer
upvoted 2 times

remz 2 years, 4 months ago

Selected Answer: A

Answer A

upvoted 1 times

🗨️ 👤 **ognamala** 2 years, 4 months ago

Selected Answer: C

As grada explained very well, the answer should be C

upvoted 1 times

🗨️ 👤 **ognamala** 2 years, 4 months ago

Actually, ignore this answer - C is retrieving from the change feed not the conflicts feed

upvoted 1 times

🗨️ 👤 **ognamala** 2 years, 4 months ago

Correct answer is A

upvoted 1 times

🗨️ 👤 **mybiai** 2 years, 5 months ago

Selected Answer: B

The Azure Cosmos DB Trigger uses the Azure Cosmos DB Change Feed to listen for inserts and updates across partitions. The change feed publishes inserts and updates, not deletions.

upvoted 2 times

🗨️ 👤 **avocacao** 2 years, 5 months ago

Answer should be A. Need to read from conflict feed but there is no trigger mechanism for the conflict feed in Cosmos DB.

upvoted 4 times

🗨️ 👤 **grada** 2 years, 5 months ago

The correct answer is C, only change feed triggers are available, and conflict feed needs to be manually queried from a timer-triggered function.

Source: <https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-cosmosdb-v2-trigger?tabs=in-process%2Cextensionv4&pivots=programming-language-csharp#attributes>

Nothing conflict-feed-related there, only change-feed-related. Conflict feed needs to be manually triggered like this: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-manage-conflicts?tabs=dotnetv3%2Capi-async%2Casync#read-from-conflict-feed>

upvoted 2 times

🗨️ 👤 **xRiot007** 12 months ago

You can use a timer trigger as well.

upvoted 1 times

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:

- ⇒ Ensure that the report can run as quickly as possible.
- ⇒ 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.

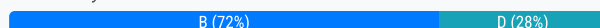
Suggested Answer: B

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>

Community vote distribution



grada Highly Voted 2 years, 5 months ago

Selected Answer: B

After processing items in the change feed, you can build a materialized view and persist aggregated values back in Azure Cosmos DB. If you're using Azure Cosmos DB to build a game, you can, for example, use change feed to implement real-time leaderboards based on scores from completed games.

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

upvoted 10 times

YellowSky002 Most Recent 5 months, 1 week ago

Selected Answer: B

For most scenarios, using the change feed to populate aggregates in Cosmos DB is the more RU-efficient approach. It allows you to process changes incrementally, update aggregates in real-time, and avoid the overhead of full scans. However, consider your specific data volume, change frequency, and aggregate complexity to choose the most appropriate strategy.

upvoted 2 times

Blubb1860 1 year ago

Selected Answer: B

B is correct, no additional SQL queries necessary

upvoted 1 times

  **comoon** 1 year, 4 months ago

Selected Answer: D

The answer is D

upvoted 1 times

  **azuredemo2022three** 1 year, 6 months ago

Selected Answer: B

upvoted 2 times



  **[Removed]** 1 year, 8 months ago

Selected Answer: B

Option D suggests populating the aggregates by using SQL queries that run daily. While this may reduce the RU consumption during querying, it may not necessarily minimize RU consumption overall. Additionally, this approach may result in stale data since the aggregates are only updated once a day.

The best approach to minimize RU consumption and ensure the report runs as quickly as possible is to use the change feed to populate the aggregates in real-time, as suggested in option B. This way, the aggregates are always up-to-date, and the report can be generated quickly and with minimal RU consumption.



upvoted 3 times

  **BOT_123** 1 year, 11 months ago

Selected Answer: B

B Is Correct

upvoted 2 times

  **TimSss** 2 years, 3 months ago

Selected Answer: D

I would go with D, we need to minimize RU, so we run the job daily which should be fine for these types of reports (using monthly data). Using the change feed costs much more RU.

upvoted 4 times

  **remz** 2 years, 4 months ago

Selected Answer: B

B Is Correct

upvoted 2 times

  **Gall** 2 years, 4 months ago

Selected Answer: D

I would go with D, as we need to reduce RU. An additional container writes costs RU and the change-feed will be updated every time a new item appears.

upvoted 2 times

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:

- ⇒ A function named Fn1 that reads the change feed of cn1
- ⇒ A function named Fn2 that reads the change feed of cn2
- ⇒ A function named Fn3 that reads the change feed of cn3

You perform the following actions:

- ⇒ Delete an item named item1 from cn1.
- ⇒ Update an item named item2 in cn2.
- ⇒ 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"/>

Suggested 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>

  **basiltomato** Highly Voted 1 year, 4 months ago

Correct -

N - <https://learn.microsoft.com/en-us/azure/cosmos-db/change-feed#features-of-change-feed>

N - <https://stackoverflow.com/questions/68409298/how-to-tell-the-difference-between-insert-and-update-in-cosmos-db-change-feed>

Y - <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>

upvoted 8 times

  **azuredemo2022three** Most Recent 1 year ago

The correct answer is NYY.

Explanation:



Fn1 will receive item1 from the change feed: No (The item was deleted from cn1, so it will not be received by Fn1.)

Fn2 checks the _etag of item2 to see whether the item is an update or an insert: Yes (Fn2 will be able to determine whether item2 is an update or an insert based on its _etag value.)

Fn3 will receive item3 from the change feed: Yes (The update of the time to live (TTL) for item3 in cn3 will trigger a change event in the change feed, and Fn3 will receive it.)

Therefore, the correct answer is NYY.

upvoted 1 times

  **azuredemo2022three** 1 year ago

My Mistake

Answer is NNY

upvoted 6 times

HOTSPOT -

You configure Azure Cognitive Search to index a container in an Azure Cosmos DB Core (SQL) API account as shown in the following exhibit.

+ Add field + Add subfield Delete								
Field Name	Type	Retrievable	Filterable	Sortable	Facetable	Searchable	Analyzer	Suggester
id	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
name	Edm.String	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Standard - Lucene	...
▼headquarters	Edm.ComplexType							...
country	Edm.String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
iso	Edm.String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
employees	Edm.Int32	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			...
🔑 rid	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...

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.

Hot Area:

Answer Area

The [answer choice] field is limited to exact match comparisons

▼

country

id

name

The [answer choice] field is hidden form the search results

▼

country

id

name

Suggested Answer:

Answer Area

The [answer choice] field is limited to exact match comparisons

▼

country

id

name

The [answer choice] field is hidden form the search results

▼

country

id

name

Box 1: country -

The country field is filterable.

Note: filterable: Indicates whether to enable the field to be referenced in \$filter queries. Filterable differs from searchable in how strings are handled. Fields of type

Edm.String or Collection(Edm.String) that are filterable do not undergo lexical analysis, so comparisons are for exact matches only.

Box 2: name -

The name field is not Retrievable.

Retrievable: Indicates whether the field can be returned in a search result. Set this attribute to false if you want to use a field (for example, margin) as a filter, sorting, or scoring mechanism but do not want the field to be visible to the end user.

Note: searchable: Indicates whether the field is full-text searchable and can be referenced in search queries.

Reference:

<https://docs.microsoft.com/en-us/rest/api/searchservice/create-index>

  **dtodorov**  9 months, 2 weeks ago

Yes, correct, Country & Name



upvoted 5 times

  **WimTS**  2 months, 2 weeks ago

Country is Filterable => exact match

Name is not Retrievable => cannot be returned in search result

upvoted 1 times

  **Bviljoen** 8 months, 3 weeks ago

<https://learn.microsoft.com/en-us/azure/search/search-what-is-an-index#field-attributes>

"filterable" - Referenced in \$filter queries. Filterable fields of type Edm.String or Collection(Edm.String) don't undergo word-breaking, so comparisons are for exact matches only. For example, if you set such a field f to "sunny day", \$filter=f eq 'sunny' finds no matches, but \$filter=f eq 'sunny day' will.

"retrievable" - Determines whether the field can be returned in a search result. This is useful when you want to use a field (such as profit margin) as a filter, sorting, or scoring mechanism, but don't want the field to be visible to the end user. This attribute must be true for key fields.

Yes County and I believe Id would be more correct as a key field.

upvoted 4 times

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 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 Synapse pipeline that uses Azure Cosmos DB Core (SQL) API as the input and Azure Blob Storage as the output.

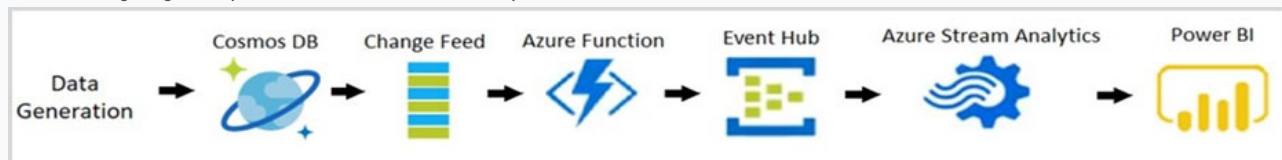
Does this meet the goal?

- A. Yes
- B. No

Suggested 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.

The following diagram represents the data flow and components involved in the solution:



Reference:

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

Community vote distribution

A (100%)

matejka 3 months, 3 weeks ago

Selected Answer: B

To make the contents of container1 available as reference data for an Azure Stream Analytics job, you should directly connect Azure Stream Analytics to Azure Cosmos DB as a reference data source. Utilizing Azure functions and Azure event hubs adds unnecessary complexity for this specific requirement. Directly connecting Azure Stream Analytics to Azure Cosmos DB ensures real-time data integration without the need for intermediate steps.

upvoted 1 times

[Removed] 7 months, 3 weeks ago

This is a series of questions with multiple scenarios. Correct options are:

1. You create an Azure Synapse pipeline that uses Azure Cosmos DB Core (SQL) API as the input and Azure Blob Storage as the output.
2. 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.

All other options are incorrect.

upvoted 2 times

3a0b61c 9 months, 2 weeks ago

Selected Answer: A

<https://learn.microsoft.com/en-us/azure/stream-analytics/stream-analytics-use-reference-data#example>

Stream Analytics supports Azure Blob Storage, Azure Data Lake Storage Gen2, and Azure SQL Database as the storage layer for reference data.

I think the solution of outputting to Azure Blob Storage is yes, and the rest is no.

upvoted 3 times

XiangRongChang 1 year, 5 months ago

According to <https://learn.microsoft.com/en-us/azure/stream-analytics/stream-analytics-quick-create-portal>, can the answer be YES.

upvoted 2 times

SwePha 1 year, 1 month ago

Probably not as I do not see CosmosDB as an input for stream analytics job

<https://learn.microsoft.com/en-us/azure/stream-analytics/stream-analytics-add-inputs>

upvoted 2 times

  **azuredemo2022three** 1 year, 6 months ago

Given answer is correct

upvoted 4 times

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 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?

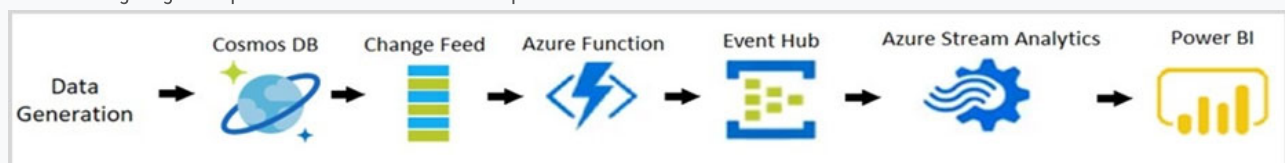
A. Yes

B. No

Suggested 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.

The following diagram represents the data flow and components involved in the solution:



Reference:

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

Community vote distribution

A (100%)

🗳️ [Removed] 7 months, 3 weeks ago

This is a series of questions with multiple scenarios. Correct options are:

1. You create an Azure Synapse pipeline that uses Azure Cosmos DB Core (SQL) API as the input and Azure Blob Storage as the output.
2. 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.

All other options are incorrect.

upvoted 1 times

🗳️ 3a0b61c 9 months, 2 weeks ago

Selected Answer: A

<https://learn.microsoft.com/en-us/azure/stream-analytics/stream-analytics-use-reference-data#example>

Stream Analytics supports Azure Blob Storage, Azure Data Lake Storage Gen2, and Azure SQL Database as the storage layer for reference data.

I think the solution of outputting to Azure Blob Storage is yes, and the rest is no.

upvoted 1 times

🗳️ SwePha 1 year, 1 month ago

Selected Answer: A

<https://learn.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities?tabs=data-factory#data-movement-activities>

Azure Data Factory supports CosmosDB as a source and Blob storage as the sink.

upvoted 4 times

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 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 function that uses Azure Cosmos DB Core (SQL) API change feed as a trigger and Azure event hub as the output. Does this meet the goal?

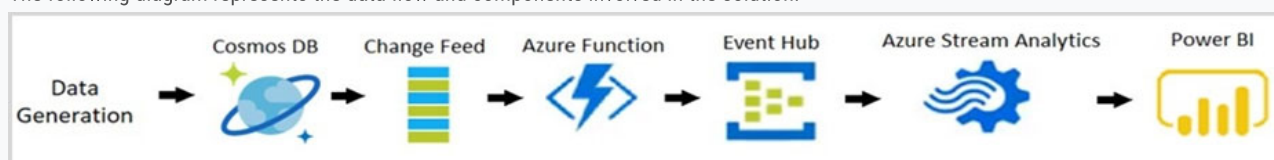
A. Yes

B. No

Suggested Answer: A

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.

The following diagram represents the data flow and components involved in the solution:



Reference:

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

Community vote distribution

B (100%)

Flammkuchen Highly Voted 2 years, 3 months ago

I don't think that Y is correct. While event hub would be a great input source for Stream Analytics *Streaming* Data, it is not supported as input for Stream Analytics *Reference* Data. <https://learn.microsoft.com/en-us/azure/stream-analytics/stream-analytics-add-inputs>.

Using Data Factory is the recommended solution for reference data in the Microsoft docs: <https://learn.microsoft.com/en-us/azure/stream-analytics/stream-analytics-use-reference-data>

upvoted 9 times

XiangRongChang 1 year, 5 months ago

So, why question #6 is no? The question before this one.

upvoted 1 times

xRiot007 12 months ago

Question #6 is wrong. For real-time data, you use an Event Hub. For reference data, you use Data Factory

upvoted 2 times

WimTS Most Recent 2 months, 2 weeks ago

Selected Answer: B

For reference data, only Blob, Data Lake Storage Gen2 and SQL are supported

For streaming data only Event Hubs, IoT Hub, Blob Storage and Data Lake Storage Gen2

upvoted 1 times

azuredemo2022three 1 year, 6 months ago

Selected Answer: B

Answer

upvoted 1 times