aws CERTIFIED Cloud Practitioner

aws CERTIFIED Developer Associate

aws CERTIFIED SysOps Administrator Associate

aws CERTIFIED Solutions Architect Associate

review

review questions

DynamoDB

V1.01

Course title

BackSpace Academy
AWS Certified Associate

BackSpace

This "learning by quizzes" exercise will be based upon the course videos and the following reference material:

Section: What is Amazon DynamoDB?

Reference: Amazon DynamoDB Developer Guide

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html

Question

The DynamoDB data model concepts include:

- Tables
- Items
- Attribute

Answers

A. True
B. False

A
The following are the basic DynamoDB components:

Tables – Similar to other database systems, DynamoDB stores data in tables. A table is a collection of data. For example, see the example table called People that you could use to store personal contact information about friends, family, or anyone else of interest. You could also have a Cars table to store information about vehicles that people drive.

Items – Each table contains zero or more items. An item is a group of attributes that is uniquely identifiable among all of the other items. In a People table, each item represents a person. For a Cars table, each item represents one vehicle. Items in DynamoDB are similar in many ways to rows, records, or tuples in other database systems. In DynamoDB, there is no limit to the number of items you can store in a table.

Attributes – Each item is composed of one or more attributes. An attribute is a fundamental data element, something that does not need to be broken down any further. For example, an item in a People table contains attributes called PersonID, LastName, FirstName, and so on. For a Department table, an item might have attributes such as DepartmentID, Name,Manager, and so on. Attributes in DynamoDB are similar in many ways to fields or columns in other database systems.

See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents.html

Question

In Amazon DynamoDB, a database is a collection of items. An item is a collection of tables and each table is a collection of attributes.

Answers

A. True
B. False

B

A database is a collection of tables. An table is a collection of items and each item is a collection of attributes.

Question

DynamoDB supports cross-table joins.

Answers

A. True
B. False

B
DynamoDB is a simple NoSQL database, not a relational database. It does not support joins.

If you want to perform join operations it needs to be done with an external application outside of DynamoDB. Amazon DynamoDB is integrated with Apache Hive, a data warehousing application that runs on Amazon EMR. Hive can read and write data in DynamoDB tables, allowing you to:

- Query live DynamoDB data using a SQL-like language (HiveQL).
- Copy data from a DynamoDB table to an Amazon S3 bucket, and vice-versa.
- Copy data from a DynamoDB table into Hadoop Distributed File System (HDFS), and vice-versa.
- Perform join operations on DynamoDB tables.

Question

Amazon DynamoDB supports the following data types:

A. Scalar types – Number, String, Binary, Boolean, and Null.
B. Multi-valued types – String Set, Number Set, and Binary Set.
C. Document types – List and Map.

Answers

A. True
B. False

A
See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.NamingRulesDataTypes.html#HowItWorks.DataTypes

Question

DynamoDB supports both eventually consistent and strongly consistent read options.

Answers

A. True
B. False

A
DynamoDB uses eventually consistent reads, unless you specify otherwise. Read operations (such as GetItem, Query, and Scan) provide a ConsistentRead parameter. If you set this parameter to true, DynamoDB uses strongly consistent reads during the operation.

See: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadConsistency.html

This "learning by quizzes" exercise will be based upon the course videos and the following reference material:

Section: Working with DynamoDB

Reference: Amazon DynamoDB Developer Guide

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithDynamo.html

Question

I need to create a total of 10 DynamoDB secondary indexes including 5 global secondary indexes. Is this possible?

Answers

A. True
B. False

A
You can create up to 5 global secondary indexes and up to 5 local secondary indexes per table.

Question

A DynamoDB global secondary index has the same partition (hash) key as the table, but a different sort (range) key.

Answers

A. True
B. False

B
Every global secondary index must have a partition key, and can have an optional sort key. The index key schema can be different from the base table schema; you could have a table with a simple primary key (partition key), and create a global secondary index with a composite primary key (partition key and sort key) — or vice-versa.

See: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GSI.html

Question

DynamoDB supports types that represent number sets, string sets and binary sets.

Answers

A. True
B. False

A
DynamoDB supports types that represent sets of Number, String, or Binary values. All of the elements within a set must be of the same type. For example, an attribute of type Number Set can only contain numbers; String Set can only contain strings; and so on.

There is no limit on the number of values in a set, as long as the item containing the values fits within the DynamoDB item size limit (400 KB).

Each value within a set must be unique. The order of the values within a set is not preserved; therefore, your applications must not rely on any particular order of elements within the set. Finally, DynamoDB does not support empty sets.

See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.NamingRulesDataTypes.html#HowItWorks.DataTypes

Question

DynamoDB provides operations to create, update and delete tables.

Answers

A. True
B. False

A
See: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithTables.Basics.html

Question

The DynamoDB ___________operation allows you to update existing attribute values, add new attributes, and delete existing attributes from an item.

Answers

A. UpdateTableItem
B. UpdateItem
C. UpdateAttribute
D. UpdateDeleteAttribute
E. All of the above

B
To update an existing item in a table, you use the UpdateItem operation. You must provide the key of the item you want to update. You must also provide an update expression, indicating the attributes you want to modify and the values you want to assign to them.

An update expression specifies how UpdateItem will modify the attributes of an item—for example, setting a scalar value, or removing elements from a list or a map.

See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.UpdateExpressions.html

Question

You can query, using the query API command, only DynamoDB tables whose primary key is of hash or hash-and-range type; you can also query any secondary index on such tables.

Answers

A. True
B. False

B
You can use GetItem or PutItem to read or write an item using its primary key only. The Query API/SDK command can query without using the sort key but you can only query a table or secondary index that has a composite primary key (a partition key and a sort key). S

See: http://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_Query.html

Question

Query is the most efficient way to retrieve items from a DynamoDB table or a secondary index.

Answers

A. True
B. False

A

The Query operation finds items based on primary key values. A Scan operation reads every item in a table or a secondary index. By default, a Scan operation returns all of the data attributes for every item in the table or index.

Question

DynamoDB supports a __________ feature that lets you specify a condition when updating an item. DynamoDB writes the item only if the specified condition is met; otherwise it returns an error.

Answers

A. Strongly Consistent Write
B. Conditional Write
C. Eventually Consistent Write
D. None of the above

B
By default, the DynamoDB write operations (PutItem, UpdateItem, DeleteItem) are unconditional: each of these operations will overwrite an existing item that has the specified primary key.

DynamoDB optionally supports conditional writes for these operations. A conditional write will succeed only if the item attributes meet one or more expected conditions. Otherwise, it returns an error. Conditional writes are helpful in many situations. For example, you might want a PutItem operation to succeed only if there is not already an item with the same primary key. Or you could prevent an UpdateItem operation from modifying an item if one of its attributes has a certain value.

Conditional writes are helpful in cases where multiple users attempt to modify the same item.

See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithItems.html#WorkingWithItems.ConditionalUpdate

Question

A DynamoDB unit of write capacity represents one write per second for items as large as 4 KB.

Answers

A. True
B. False

B
You specify throughput capacity in terms of read capacity units and write capacity units:

One read capacity unit represents one strongly consistent read per second, or two eventually consistent reads per second, for an item up to 4 KB in size. If you need to read an item that is larger than 4 KB, DynamoDB will need to consume additional read capacity units. The total number of read capacity units required depends on the item size, and whether you want an eventually consistent or strongly consistent read.

One write capacity unit represents one write per second for an item up to 1 KB in size. If you need to write an item that is larger than 1 KB, DynamoDB will need to consume additional write capacity units. The total number of write capacity units required depends on the item size.

See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ProvisionedThroughput.html

Question

A DynamoDB unit of read capacity represents one strongly consistent read per second (or two eventually consistent reads per second) for items as large as 4 KB.

Answers

A. True
B. False

A
See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ProvisionedThroughput.html

Question

Calculate the number of DynamoDB read operations (read capacity units) for an item of 13 KB:

Answers

A. 3
B. 4
C. 2
D. 6

B
13/4 = 3.25
4 (3.25 rounded up) read capacity units required

Question

If your DynamoDB items are 5 KB and you want to read 80 items per second from your table, then you need to provision ___ read capacity units for strong consistency.

Answers

A. 40
B. 80
C. 120
D. 160

D
One read capacity unit represents one strongly consistent read per second, or two eventually consistent reads per second, for an item up to 4 KB in size.

5 KB requires two strongly consistent (4 KB) read capacity units
2 x 80 = 160

See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ProvisionedThroughput.html

Question

If your DynamoDB items are 5 KB and you want to read 80 items per second from your table, then you need to provision ___ read capacity units for eventual consistency.

Answers

A. 40
B. 80
C. 120
D. 160

B
One read capacity unit represents one strongly consistent read per second, or two eventually consistent reads per second, for an item up to 4 KB in size.

5 KB requires one eventually consistent read capacity units
1 x 80 = 80

See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ProvisionedThroughput.html

Question

if your DynamoDB items are 3.5 KB and you want to do 20 writes per second, then you would need to provision _____ write capacity units.

Answers

A. 20
B. 40
C. 60
D. 80

D
One write capacity unit represents one write per second for an item up to 1 KB in size.
4 capacity units required for 3.5 KB.
4x20 = 80

See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ProvisionedThroughput.html

Question

The following are valid DynamoDB table names:

- Id
- cool.table.name
- cool-table-name
- cool:table:name

Answers

A. True
B. False

B
cool.table.name
cool-table-name

For table and secondary index names, allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (dash), and '.' (dot). Names can be between 3 and 255 characters long.

See:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.NamingRulesDataTypes.html#HowItWorks.NamingRules

Question

If you expect upcoming spikes in your workload (such as a new product launch) that will cause your throughput to exceed the current provisioned throughput for your table, you can use the DynamoDB UpdateTable operation to increase the ProvisionedThroughput value.

Answers

A. True
B. False

A
As your application data and access requirements change, you might need to adjust your table's throughput settings. If you're using DynamoDB auto scaling, the throughput settings are automatically adjusted in response to actual workloads. You can also use the UpdateTable operation to manually adjust your table's throughput capacity. You might decide to do this if you need to bulk-load data from an existing data store into your new DynamoDB table. You could create the table with a large write throughput setting and then reduce this setting after the bulk data load is complete.

See: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/ProvisionedThroughput.html

Question

The DynamoDB capacity units consumed by an operation depends on the following:

- Item size
- Read consistency (in case of a read operation)
- Provisioned IOPS

Answers

A. True
B. False

B
Item size, Read consistency (in case of a read operation).
Provisioned IOPS only makes them consumed faster.

Question

The optimal usage of a DynamoDB table's provisioned throughput depends on these factors:

- Database schema
- The primary key selection.
- The workload patterns on individual items

Answers

A. True
B. False

B
The primary key selection.The workload patterns on individual items

See: https://aws.amazon.com/blogs/database/choosing-the-right-dynamodb-partition-key/

Question

Choose the DynamoDB partition (hash) key schema for provisioned throughput efficiency:

Answers

A. Status code, where there are only a few possible status codes.
B. User ID, where the application has many users.
C. Device ID, where even if there are a lot of devices being tracked, one is by far more popular than all the others.
D. Item creation date, rounded to the nearest time period (e.g. day, hour, minute)

B
See: https://aws.amazon.com/blogs/database/choosing-the-right-dynamodb-partition-key/

Question

There is a limit of 256 KB on the DynamoDB item size.

Answers

A. True
B. False

B
The maximum item size in DynamoDB is 400 KB, which includes both attribute name binary length (UTF-8 length) and attribute value lengths (again binary length). The attribute name counts towards the size limit.

See: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Limits.html#limits-items

Question

You must provide the name of the DynamoDB table, along with the primary key of the item you want when using the GetItem operation.

Answers

A. True
B. False

A
DynamoDB provides four operations for basic create/read/update/delete (CRUD) functionality:

- PutItem – create an item.
- GetItem – read an item.
- UpdateItem – update an item.
- DeleteItem – delete an item.

Each of these operations require you to specify the primary key of the item you want to work with. For example, to read an item using GetItem, you must specify the partition key and sort key (if applicable) for that item.

See: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithItems.html

Question

To create, update, and delete items in a DynamoDB table, use the following operations:

- PutItem
- UpdateItem
- WriteItem
- DeleteItem

Answers

A. True
B. False

B
PutItem UpdateItem DeleteItem

See: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithItems.html

Question

A single DynamoDB BatchGetItem request can retrieve up to 16 MB of data, which can contain as many as 25 items.

Answers

A. True
B. False

B
BatchGetItem
A single BatchGetItem operation can retrieve a maximum of 100 items. The total size of all the items retrieved cannot exceed 16 MB.

See: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Limits.html#limits-api

Question

DynamoDB BatchWriteItem can write up to 1 MB of data, consisting of up to 100 put or delete requests.

Answers

A. True
B. False

B
A single call to BatchWriteItem can write up to 16 MB of data, which can comprise as many as 25 put or delete requests.

See: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Limits.html#limits-api

For certification practice exams please go to:

backspace.academy