

Quiz: DynamoDB

Chandra Lingam, Copyright © 2019 Cloud Wave LLC. All Rights Reserved.

1. You are developing a mobile application that customizes user experience based on logged on user. Your application needs to scale to very large number of concurrent users with consistent millisecond latency.

What backend stores can you use for storing user sessions?

- A. RDS
 - B. Redshift
 - C. ElasticSearch
 - D. DynamoDB
-
2. When you use DynamoDB Service, you need to specify:
 - A. Server Instance Class
 - B. Throughput needed
 - C. DynamoDB Engine Database Administrator User ID/Password
 - D. All the above

 3. When defining a DynamoDB table, at the minimum you need to specify:
 - A. Primary Key
 - B. Primary Key and other Attributes
 - C. All Attributes that you plan to store in the table

 4. DynamoDB data is automatically replicated within
 - A. Availability Zone
 - B. Across all Availability Zones in a Region
 - C. Across Availability Zones if deployed using multi-AZ configuration
 - D. Across Regions

 5. You have a product inventory system maintained using DynamoDB.

Anytime a new product is added or existing product is changed, you need to update ElastiCache in-memory storage with the new changes. ElastiCache Update needs to happen in exact order as it happened in DynamoDB. What are your options?

- A. Configure DynamoDB Streams to push changes to ElastiCache

- B. Periodically poll the DynamoDB tables for any new changes and update ElastiCache
 - C. Configure lambda function to poll DynamoDB streams and update ElastiCache
 - D. Configure ElastiCache to sync data from DynamoDB
6. DynamoDB read operation always returns the latest and most up-to-date data available in a table
- A. True
 - B. False
7. Which offers higher throughput in DynamoDB when reading items?
- A. Eventual Consistency
 - B. Strong Consistency
8. You are designing a game application with DynamoDB table as the backend for tracking user scores. Primary key for the table consists of GameName as hash key and UserName as sort key.

How do you ensure DynamoDB table can scale to support very large number of users?

- A. DynamoDB can dynamically scale based on throughput needed. No further action is needed
 - B. Use On-Demand Read/Write Capacity management to ensure DynamoDB can scale to required volume
 - C. Partitions may not be evenly distributed in this scheme
9. A game application records user scores in a DynamoDB tables. Primary key consists of User ID as Hash key and game title as sort key.

Would this table design handle frequent queries to find out games played by a user?

- A. DynamoDB would use primary key to quickly retrieve the matching items
 - B. DynamoDB would need to scan the entire partition to retrieve the matching items
 - C. DynamoDB would need to scan the entire table to retrieve the matching items
10. A game application records player scores by game title. Application uses DynamoDB for storing scores.

Primary key is defined using PlayerID as hash key and GameTitle as sort key.

There are no other indexes defined.

To find the top 10 high scoring players for a game:

- A. DynamoDB would use primary key to quickly retrieve the matches.
- B. DynamoDB must scan the entire table to find the answer
- C. DynamoDB must scan the entire partition that has the game details

Answers:

1. D – DynamoDB provides consistent, single-digit millisecond latency at any scale. ElastiCache can also be used for managing user sessions. It provides sub-millisecond latency to power real-time applications.
2. B - Amazon DynamoDB integrates with AWS Identity and Access Management (IAM) for fine-grained access control for users within your organization. When creating a table, simply specify how much throughput you need and the service automatically provisions required resource. It is a serverless fully managed service.
3. A – Primary Key is the only required attribute that you need to specify when creating a DynamoDB table. All other attributes are optional as tables are schemaless.
4. B – DynamoDB automatically replicates data across multiple availability zones in a region
5. C - DynamoDB Streams captures a time-ordered sequence of item-level modifications in any DynamoDB table, and stores this information in a log for up to 24 hours. Applications can access this log and view the data items as they appeared before and after they were modified, in near real time. Lambda function can poll streams periodically to capture new changes and update cache.
6. B – DynamoDB Reads offers eventual consistency by default and it can return stale data. To get the most up-to-date data, use Strong consistency
7. A - Eventual consistency maximizes read throughput. Each provisioned read-capacity-unit can support 2 Eventual consistency reads
8. C - All partitions must have approximately equal number of items for maximum throughput. Primary Key is made up of Game name (Hash) and User Name (Sort). If all users play only one or two popular games, then it can lead to over utilization of small number of partitions
9. A – Since the query is looking for games played by an user, DynamoDB can use Primary Key to locate the partition containing user data and then directly access game title played by the user.
10. B - Since there is no secondary index defined for game title and top score, it must scan the entire table. To speed-up, you can create a secondary index based on these two attributes