## What is DynamoDB?

A CLOUD GURU

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed database and supports both document and key-value data models. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, IoT, and many other applications.

## DynamoDB

A CLOUD GURU

- Stored on SSD storage
- Spread Across 3 geographically distinct data centers
- Choice of 2 consistency models:
  - Eventual Consistent Reads (Default)
  - Strongly Consistent Reads

## DynamoDB

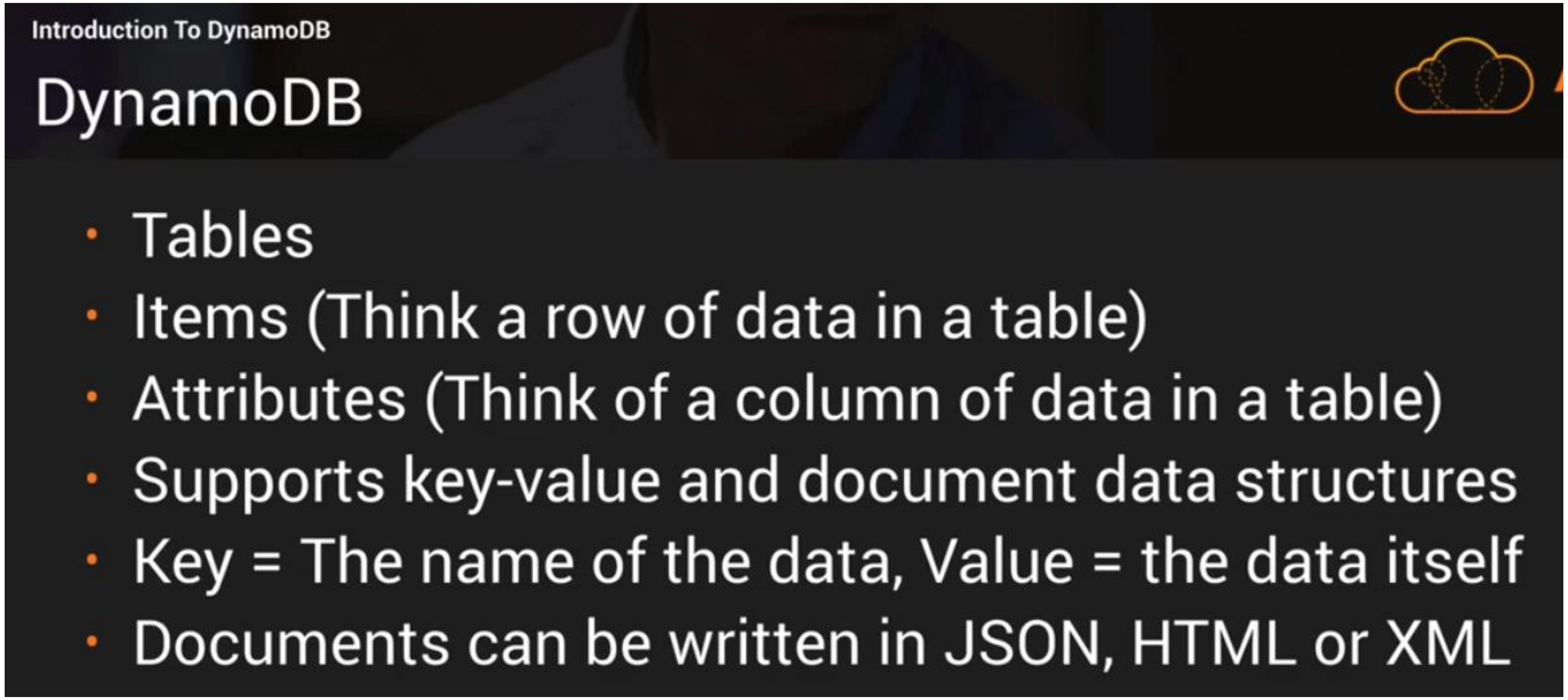
A CLOUD GURU

### Eventually Consistent Reads:

- Consistency across all copies of data is usually reached within a second. Repeating a read after a short time should return the updated data. (Best Read Performance)

### Strongly Consistent Reads:

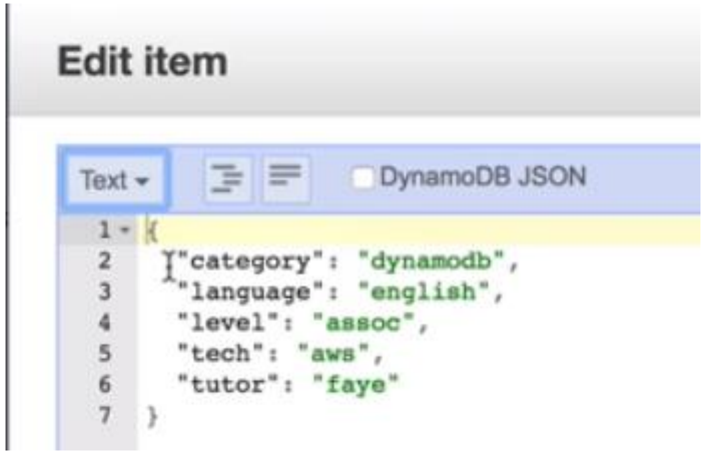
- A strongly consistent read returns a result that reflects all writes that received a successful response prior to the read.



Introduction To DynamoDB

## DynamoDB

- Tables
- Items (Think a row of data in a table)
- Attributes (Think of a column of data in a table)
- Supports key-value and document data structures
- Key = The name of the data, Value = the data itself
- Documents can be written in JSON, HTML or XML



Edit item

```
{
 "category": "dynamodb",
 "language": "english",
 "level": "assoc",
 "tech": "aws",
 "tutor": "faye"
}
```

## DynamoDB - Primary Keys

- DynamoDB stores and retrieves data based on a Primary Key
- 2 types of Primary Key:
- Partition Key - unique attribute (e.g. user ID)
- Value of the Partition key is input to an internal hash function which determines the partition or physical location on which the data is stored.
- If you are using the Partition Key as your Primary Key, then no two items can have the same Partition Key.

## DynamoDB - Primary Keys

- Composite Key (Partition Key + Sort Key) in combination
- e.g. Same user posting multiple times to a forum
  - Primary Key would a Composite Key consisting of:
  - Partition Key - User ID
  - Sort key  - Timestamp of the post
  - 2 items may have the same Partition Key, but they must have a different Sort Key
  - All items with the same Partition Key are stored together, then sorted according to the Sort Key value
  - Allows you to store multiple items with the same Partition Key

## DynamoDB - Students Table

A CLOUD GURU

```
{
"UniqueID" : 1975,                    ← Partition Key
"FirstName" : "Allan"
"Surname" : "Brown"
"Phone" : "555-2323"
}

{
"UniqueID" : 1976,                    ← Partition Key
"FirstName" : "Riad"
"Surname" : "Ramanov"
"CourseName" : "AWS_Developer_Associate"        ← Sort Key
"Address" : " {
 "Number" : "5"
 "Street" : "River Road"
               }
}
```

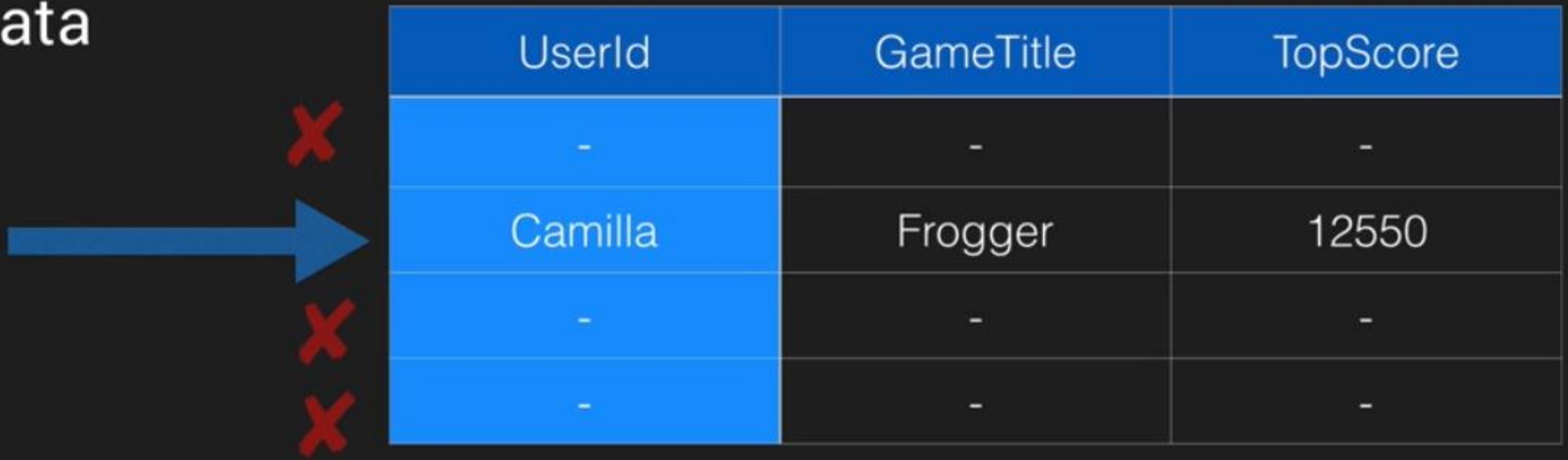
## DynamoDB Access Control

A CLOUD GURU

- Authentication and Access Control is managed using AWS IAM.
- You can create an IAM user within your AWS account which has specific permissions to access and create DynamoDB tables.
- You can create an IAM role which enables you to obtain temporary access keys which can be used to access DynamoDB.
- You can also use a special IAM Condition to restrict user access to only their own records.

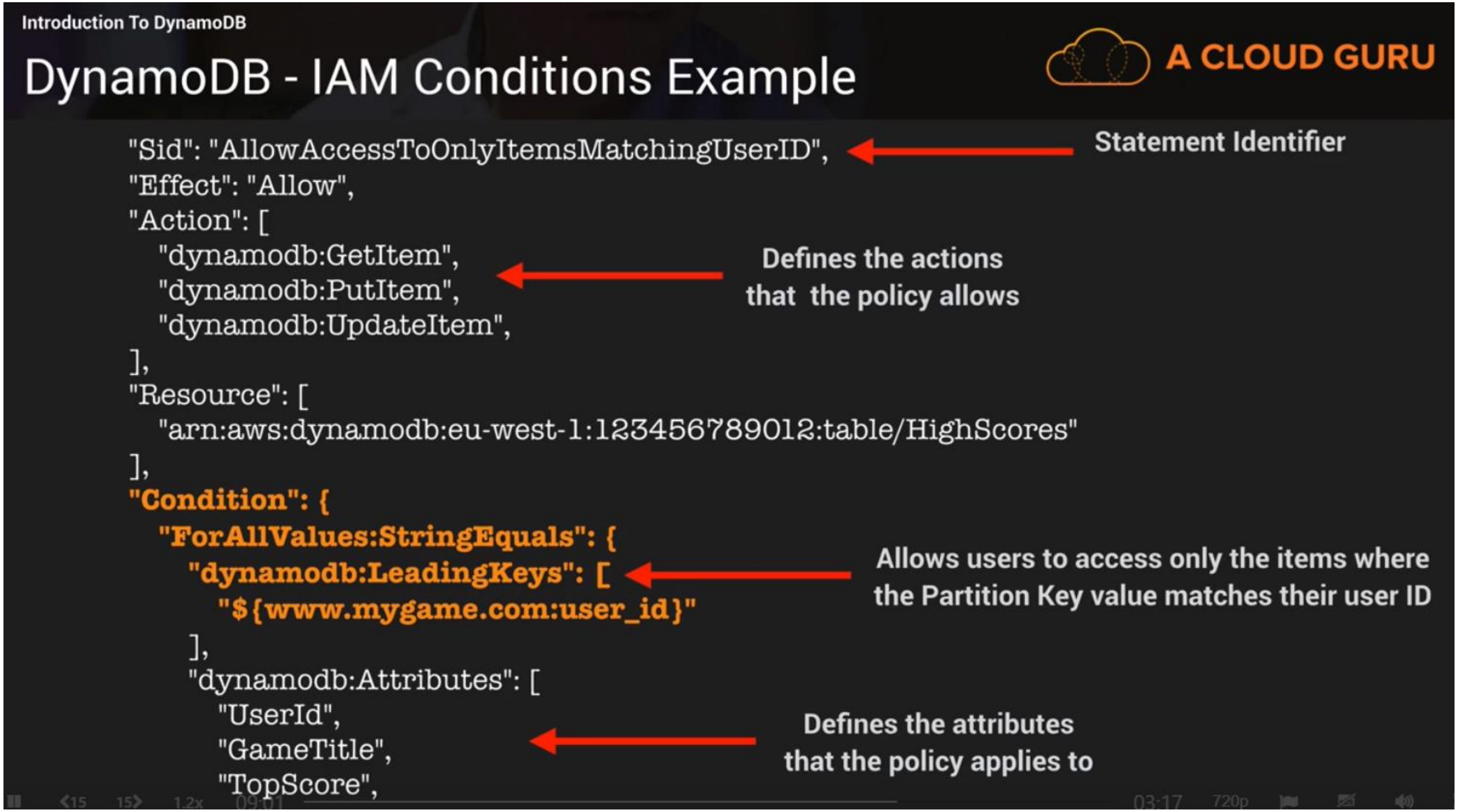
## DynamoDB - IAM Conditions Example

- Imagine a mobile gaming application with millions of users
- Users need to access the high scores for each game they are playing
- Access must be restricted to ensure they cannot view anyone else's data



| | UserId | GameTitle | TopScore |
|---|---|---|---|
| ✗ | - | - | - |
| → | Camilla | Frogger | 12550 |
| ✗ | - | - | - |
| ✗ | - | - | - |

- This can done by adding a Condition to an IAM Policy to allow access only to items where the Partition Key value matches their User ID



Introduction To DynamoDB

## DynamoDB - IAM Conditions Example

A CLOUD GURU

```
"Sid": "AllowAccessToOnlyItemsMatchingUserID",
"Effect": "Allow",
"Action": [
   "dynamodb:GetItem",
   "dynamodb:PutItem",
   "dynamodb:UpdateItem",
],
"Resource": [
   "arn:aws:dynamodb:eu-west-1:123456789012:table/HighScores"
],
"Condition": {
   "ForAllValues:StringEquals": {
      "dynamodb:LeadingKeys": [
         "${www.mygame.com:user_id}"
      ],
      "dynamodb:Attributes": [
         "UserId",
         "GameTitle",
         "TopScore",
```

Statement Identifier

Defines the actions that the policy allows

Allows users to access only the items where the Partition Key value matches their user ID

Defines the attributes that the policy applies to

## DynamoDB Exam Tips

A CLOUD GURU

- Amazon DynamoDB is a low latency NoSQL database
- Consists of Tables Items and Attributes
- Supports both document and key-value data models
- Supported document formats are JSON, HTML, XML
- 2 types of Primary Key - Partition Key and combination of Partition Key + Sort Key (Composite Key)

## DynamoDB Exam Tips

- 2 Consistency models : Strongly Consistent / Eventually Consistent

- Access is controlled using IAM policies

- Fine grained access control using IAM Condition parameter: **dynamodb:LeadingKeys** to allow users to access only the items where the partition key value matches their user ID