

Hands-On Activity: Create and Query an DynamoDB Table

About this Hands-On Activity

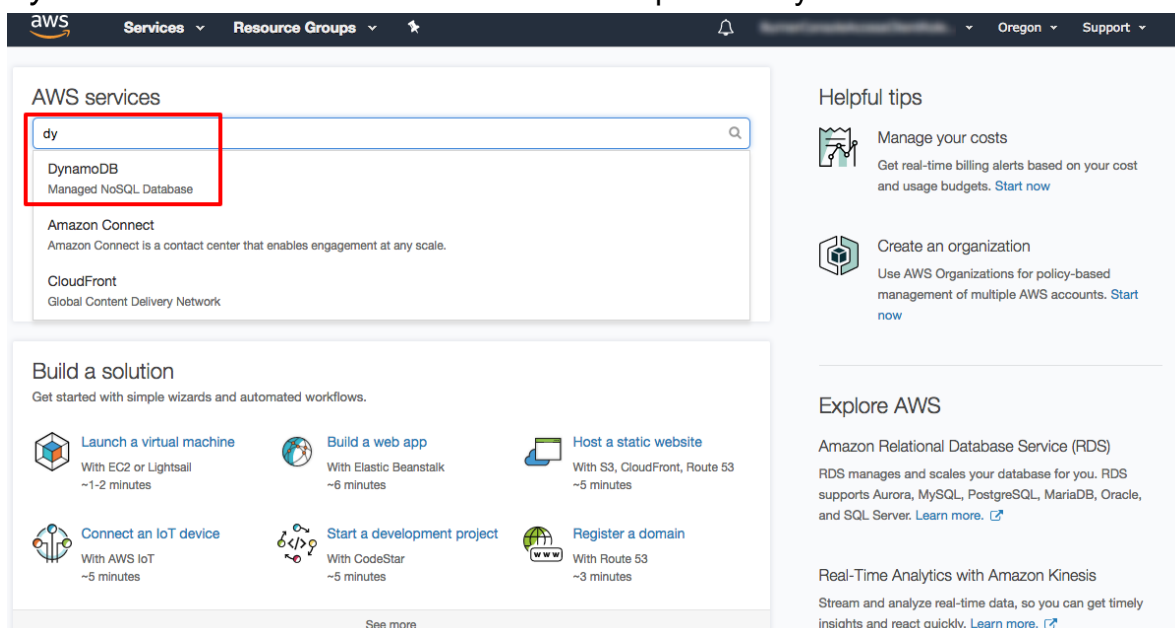
After you complete the instructions on this page, you will be able to do the following:

- Create and launch a DynamoDB table - Creating a table includes choosing a partition key and an optional sort key. Unlike what we have been working with in the RDS universe, this database will be schemaless so we won't need to specify much more than a table name and a primary key.

In this activity, you will learn how to create a simple table, add data, scan, and query the data, delete data, and delete the table by using the DynamoDB console. DynamoDB is a fully managed NoSQL database that supports both document and key-value store models. Its flexible data model, reliable performance, and automatic scaling of throughput capacity make it a great fit for mobile, web, gaming, ad tech, IoT, and many other applications.

Let's Get Started!

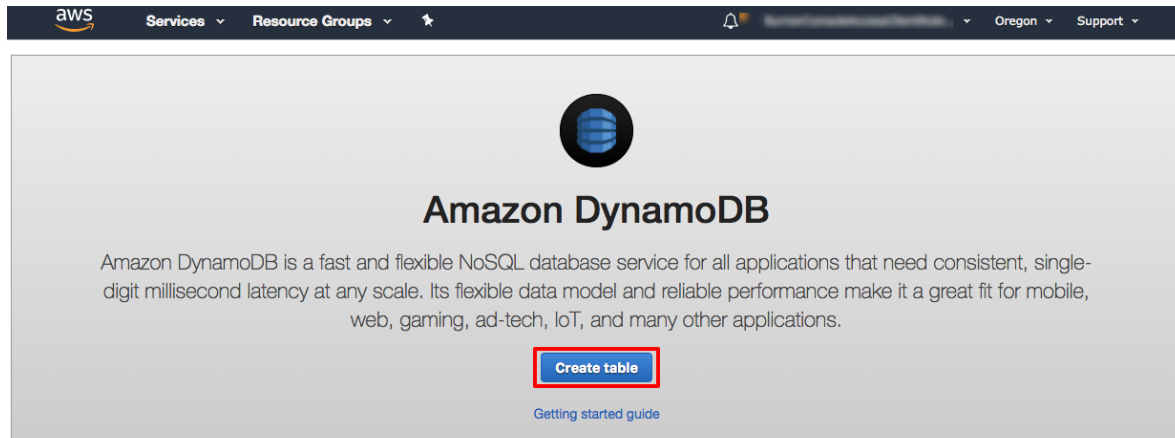
Open the AWS Console using your **AWS Educate Starter Account** so that you can keep this step-by-step guide open. When this screen loads, begin typing *DynamoDB* in the search bar and choose to open the DynamoDB console.



Step 1: Create a NoSQL Table

In this step, you will use the DynamoDB console to create a table.

a. In the DynamoDB console, choose **Create table**.



b. We will use a music library as our use case for this tutorial. In the **Table name** box, type **Music**.

c. The partition key is used to spread data across partitions for scalability. It's important to choose an attribute with a wide range of values and that is likely to have evenly distributed access patterns. Type **Artist** in the **Partition key** box.

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* Music ⓘ

Primary key* Partition key

Artist String ⓘ

☐ Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- On-Demand Backup and Restore Enabled **NEW!**

ⓘ You do not have the required role to enable Auto Scaling by default.
Please refer to [documentation](#).

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

d. Because each artist may write many songs, you can enable easy sorting with a sort key. Select the Add sort key check box. Type **songTitle** in the **Add sort key** box.

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* Music ⓘ

Primary key* Partition key

Artist String ⓘ

☒ Add sort key

SongTitle String ⓘ

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- On-Demand Backup and Restore Enabled **NEW!**

ⓘ You do not have the required role to enable Auto Scaling by default.
Please refer to [documentation](#).

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel Create

e. Next, you will enable DynamoDB auto scaling for your table.

DynamoDB auto scaling will change the read and write capacity of your table based on request volume. Using an AWS Identity and Access Management (AWS IAM) role called **DynamoDBAutoscaleRole**, DynamoDB will manage the auto scaling process on your behalf. DynamoDB creates this role for you the first time you enable auto scaling in an account.

Instruct DynamoDB to create the role by clearing the **Use default settings** check box.

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* Music ⓘ

Primary key* Partition key

Artist String ⓘ

☒ Add sort key

SongTitle String ⓘ

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ **Use default settings**

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- On-Demand Backup and Restore Enabled **NEW!**

ⓘ You do not have the required role to enable Auto Scaling by default.
Please refer to [documentation](#).

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel Create

f. Scroll down the screen past **Secondary indexes**, **Provisioned capacity**, and **Auto Scaling** to the **Create** button. We won't change these settings for the activity. In the Auto Scaling section, notice that DynamoDB will create the **DynamoDBAutoscaleRole** role for you.

Now choose, **Create**.

When the **Music** table is ready to use, it appears in the table list with a checkbox. Congratulations! You have created a NoSQL table using the DynamoDB console.

Read capacity units

Table: 5

Write capacity units

5

Estimated cost: \$2.91 / month ([Capacity calculator](#))

Auto Scaling

☒ **Read capacity**

Target utilization: 70 %

Minimum provisioned capacity: 5 units

Maximum provisioned capacity: 10000 units

☒ Apply same settings to global secondary indexes

☒ **Write capacity**

☐ Same settings as read

70 %

5 units

10000 units

☒ Apply same settings to global secondary indexes

Information Please check your IAM permissions to create new service role for enabling Auto Scaling. See [permissions](#).

IAM Role I authorize DynamoDB to scale capacity using the following role:

☒ New role: **DynamoDBAutoscaleRole**

☐ Existing role with pre-defined policies ([Instructions](#))

Role Name*

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel **Create**

Step 2: Add Data to the NoSQL Table

In this step, you will add data to your new DynamoDB table.

a. Select the **Items** tab. On the **Items** tab, choose, **Create item**.

DynamoDB

Dashboard

Tables

Backups

Reserved capacity

DAX

Dashboard

Clusters

Subnet groups

Parameter groups

Events

Create table **Delete table**

Filter by table name

Name

Music

Music **Close**

Overview **Items** Metrics Alarms Capacity Indexes Global Tables Backups Triggers More

Create item Actions

Scan: [Table] Music: Artist, SongTitle

Viewing 0 to 0 items

Scan [Table] Music: Artist, SongTitle

Add filter

Start search

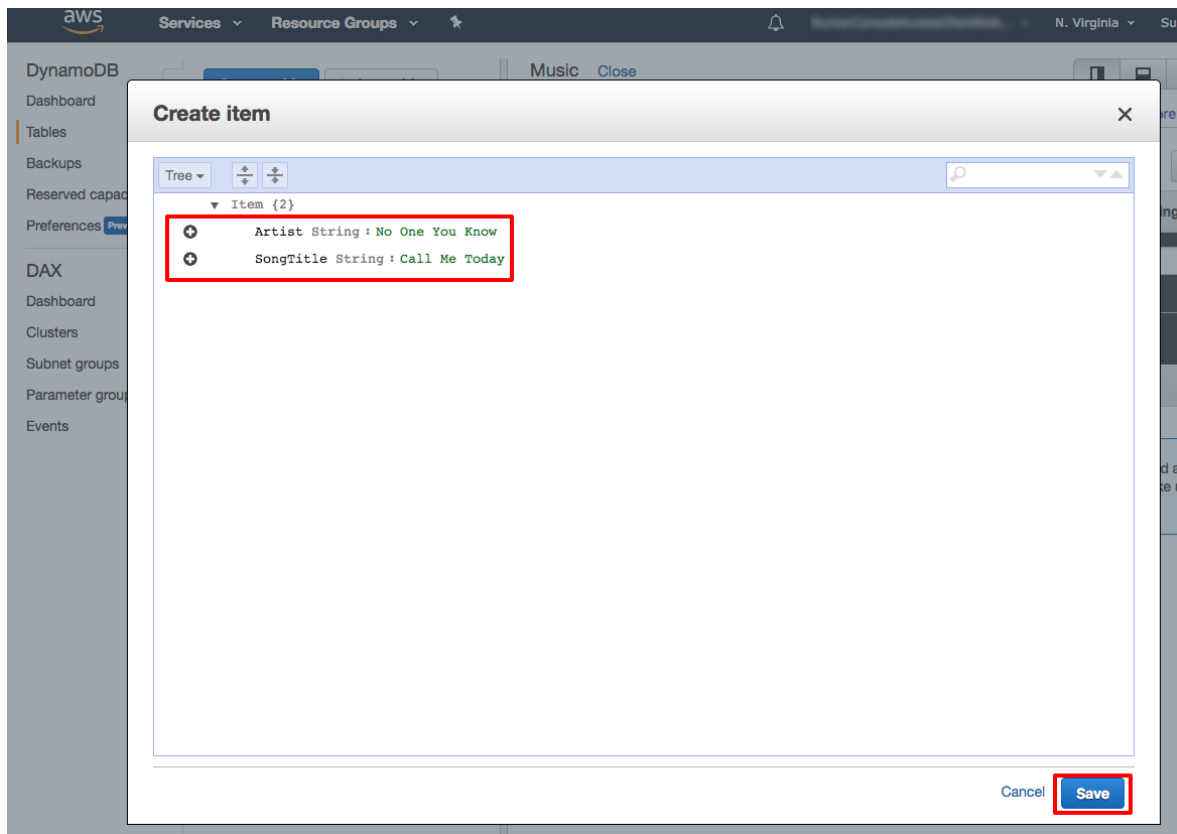
Artist SongTitle

An item consists of one or more attributes. Each attribute consists of a name, a data type, and a value. When you read or write an item, the only attributes that are required are those that make up the primary key. [More info](#)

b. In the data entry window, type the following:

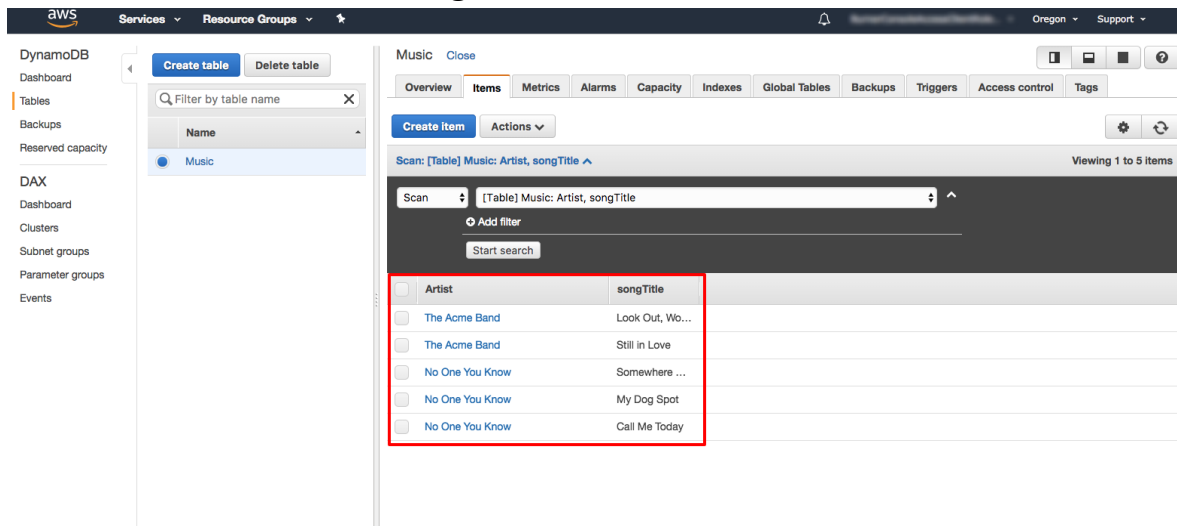
- For the **Artist** attribute, type **No One You Know**.
- For the **songTitle** attribute, type **Call Me Today**.

Choose **Save** to save the item.



c. Repeat the process to add a few more items to your Music table:

- **Artist:** No One You Know; **songTitle:** My Dog Spot
- **Artist:** No One You Know; **songTitle:** Somewhere Down The Road
- **Artist:** The Acme Band; **songTitle:** Still in Love
- **Artist:** The Acme Band; **songTitle:** Look Out, World



Step 3: Query the NoSQL Table

In this step, you will search for data in the table using query operations. In DynamoDB, query operations are efficient and use keys to find data. Scan operations traverse the entire table.

a. In the drop-down list in the dark gray banner above the items, change **Scan** to **Query**.

The screenshot shows the AWS Management Console for a DynamoDB table named 'Music'. The 'Query' tab is active, and the query is configured to search by the 'Artist' partition key and 'songTitle' sort key. The 'Start search' button is highlighted with a red box. The results table shows the following data:

Artist	songTitle
No One You Know	Call Me Today
No One You Know	My Dog Spot
No One You Know	Somewhere ...
The Acme Band	Look Out, Wo...

b. You can use the console to query the **Music** table in various ways. For your first query, do the following:

- In the **Artist** box, type **No One You Know**, and choose **Start search**. All songs performed by **No One You Know** are displayed.

Try another query:

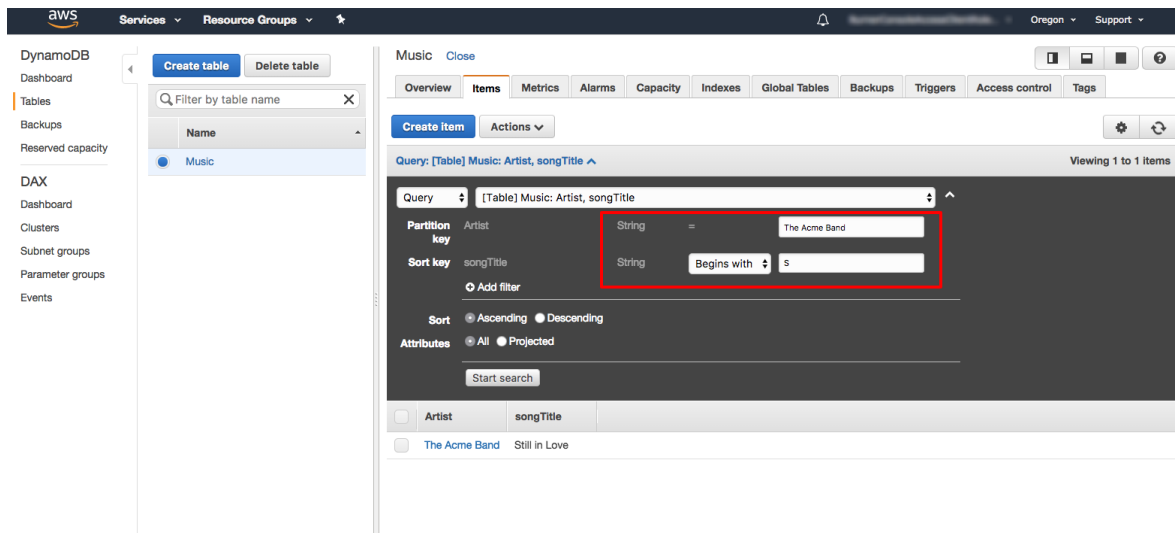
- In the **Artist** box, type **The Acme Band**, and choose **Start search**. All songs performed by **The Acme Band** are displayed.

The screenshot shows the AWS Management Console for a DynamoDB table named 'Music'. The 'Query' tab is active, and the query is configured to search by the 'Artist' partition key and 'songTitle' sort key. The 'Start search' button is highlighted with a red box. The results table shows the following data:

Artist	songTitle
No One You Know	Call Me Today
No One You Know	My Dog Spot
No One You Know	Somewhere ...

c. Try another query, but this time narrow down the search results:

- In the **Artist** box, type **The Acme Band**.
- In the **songTitle** box, select **Begins with** from the drop-down list and type **S**.
- Choose **Start search**. Only **"Still in Love"** performed by **The Acme Band** is displayed.

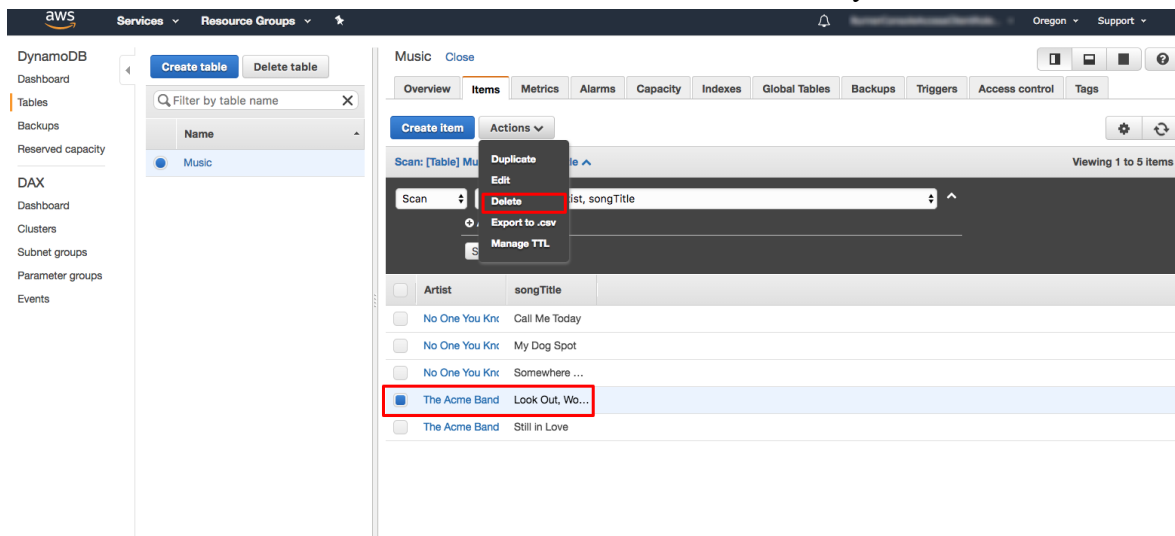


Step 4: Delete an Existing Item

In this step, you will delete an item from your DynamoDB table.

a. Change the **Query** drop-down list back to **Scan**.

Select the checkbox next to **The Acme Band**. In the **Actions** drop-down list, choose **Delete**. You will be asked whether to delete the item. Choose **Delete** and your item is deleted.

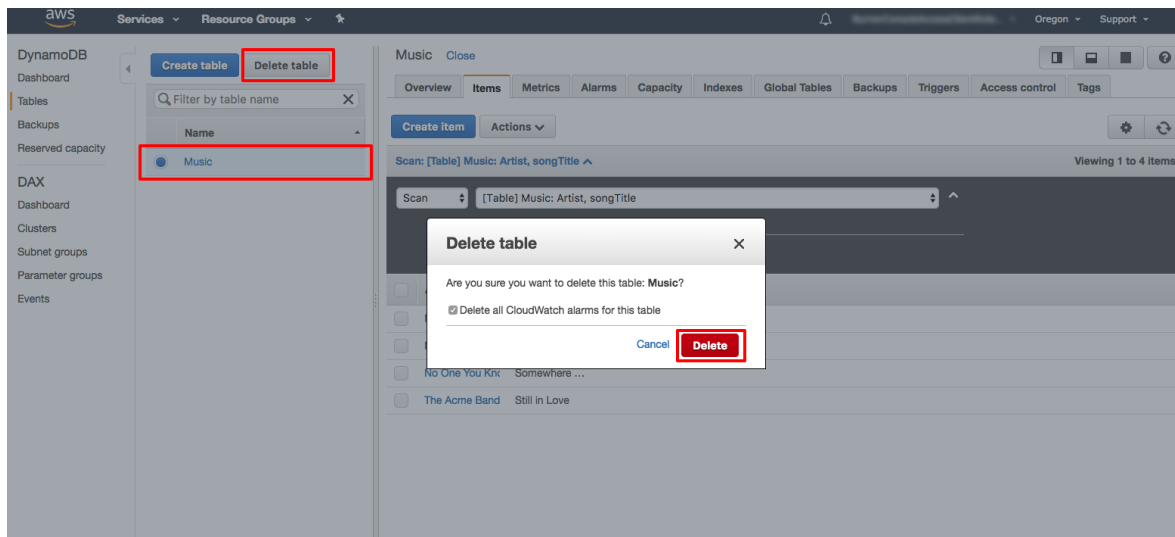


Step 5: Delete a NoSQL Table

In this step, you will delete your DynamoDB table.

a. You can easily delete a table from the DynamoDB console. It is a best practice to delete tables you are no longer using so that you don't keep getting charged for them.

- In the DynamoDB console, choose the option next to the Music table and then choose Delete table.
- In the confirmation dialog box, choose Delete.



Congratulations!

You have created your first DynamoDB table, added items to your table, and then queried the table to find the items you wanted. You also learned how to visually manage your DynamoDB tables and items through the AWS Management Console. DynamoDB is a great fit for mobile, web, gaming, ad tech, and IoT applications where scalability, throughput, and reliable performance are key considerations.