

---

# Introduction to Amazon DynamoDB

- Lab Overview
- Start lab
- Task 1: Create a New Table
- Task 2: Add Data
- Task 3: Modify an Existing Item
- Task 4: Query the Table
- Task 5: Delete the Table
- End lab
- Conclusion

## Lab Overview

**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.

In this lab, you will create a table in Amazon DynamoDB to store information about a music library. You will then query the music library and, finally, delete the DynamoDB table.

## Objectives

After completing this lab, you will be able to:

- Create an Amazon DynamoDB table

- Enter data into an Amazon DynamoDB table
- Query an Amazon DynamoDB table
- Delete an Amazon DynamoDB table

## Icon key

Various icons are used throughout this lab to call attention to certain aspects of the guide. The following list explains the purpose for each one:

- **Command:** A command that you must run.
- **Expected output:** A sample output that you can use to verify the output of a command or edited file.
- **Note:** A note, tip, or important guidance.
- **Additional information:** Where to find more information.

## Task 1: Create a New Table

In this task, you will create a new table in DynamoDB named *Music*. Each table requires a Primary Key that is used to partition data across DynamoDB servers. A table can also have a Sort Key. The combination of Primary Key and Sort Key uniquely identifies each item in a DynamoDB table.

3. From the **AWS Management Console**, use the **AWS search bar** to search for  and then choose the service from the list of results in a new browser tab.
4. Choose the **Create table** button and configure the following options:

### Table details

- **Table name:**

- **Partition key:**
- **Sort key - \*optional\*:**

Your table will use default settings for indexes and provisioned capacity.

5. Choose **Create table**.

**Expected output:** Initial message.

Creating the Music table. It will be available for use shortly.

6. Wait for your table to be created and its status to be **Active**.

## Task 2: Add Data

In this task, you will add data to the *Music* table. A **table** is a collection of data on a particular topic.

Each table contains multiple **items**. An item is a group of attributes that is uniquely identifiable among all of the other items. Items in DynamoDB are similar in many ways to rows in other database systems. In DynamoDB, there is no limit to the number of items you can store in a table.

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 *Music* table contains attributes such as Song and Artist. Attributes in DynamoDB are similar columns in other database systems, but each item (row) can have different attributes (columns).

When you write an item to a DynamoDB table, only the Primary Key and Sort Key (if used) are required. Other than these fields, the table does not require a schema. This means that you can add attributes to one item that may be different to the attributes on other items.

7. In the left navigation pane, choose **Explore items**.
8. Choose the **Music** table.
9. Click on **Explore table items**
10. Choose **Create item** .

11. For **Artist** Value, enter:

12. For **Song** Value, enter:

These are the only required attributes, but you will now add additional attributes.

12. To create an additional attribute, choose **Add new attribute** .
13. In the drop-down list, select **String**.

A new attribute row will be added.

14. For the new attribute, enter:
  - In **Attribute name**, enter:
  - In **Value**, enter:
15. Create another new attribute by choosing **Add new attribute** .

16. In the drop-down list select **Number**.

A new *number* attribute will be added.

17. For the new attribute, enter:
  - In **Attribute name**, enter:
  - In **Value**, enter:
18. Choose **Create item** to store the new Item with its four attributes.

The item will appear in the console.

19. Create a second item, using these attributes:

Attribute Name	Attribute Type	Attribute Value
<i>Artist</i>	String	John Lennon
<i>Song</i>	String	Imagine
<i>Album</i>	String	Imagine
<i>Year</i>	Number	1971
<i>Genre</i>	String	Soft rock

Note that this item has an additional attribute called *genre*. This is an example of each item being capable of having different attributes without having to pre-define a table schema.

20. Create a third item, using these attributes:

Attribute Name	Attribute Type	Attribute Value
<i>Artist</i>	String	Psy
<i>Song</i>	String	Gangnam Style
<i>Album</i>	String	Psy 6 (Six Rules), Part 1
<i>Year</i>	Number	2011

<i>LengthSeconds</i>	Number	219
----------------------	--------	-----

Once again, this item has a new *LengthSeconds* attribute identifying the length of the song. This demonstrates the flexibility of a NoSQL database.

There are also faster ways to load data into DynamoDB, such as using AWS Data Pipeline, programmatically loading data or using one of the free tools available on the Internet.

## Task 3: Modify an Existing Item

In this task, you will modify an existing item.

21. Choose **Psy**.
22. Change the **Year** from *2011* to *2012*.
23. Choose **Save changes**

The item is now updated.

## Task 4: Query the Table

There are two ways to query a DynamoDB table: *Query* and *Scan*.

A **query** operation finds items based on Primary Key and optionally Sort Key. It is fully indexed, so it runs very fast.

24. In the left navigation pane, choose **Explore items**.

25. Choose the **Music** table.

26. Expand **Scan/Query items** to query or scan items.

27. Choose **Query**.

Fields for the Partition Key (which is the same as Primary Key) and Sort Key are now displayed.

28. Enter these details:

- **Artist (Partition key):**
- **Song (Sort key):**

29. Choose **Run**.

The song quickly appears in the list. A *query* is the most efficient way to retrieve data from a DynamoDB table.

Alternatively, you can *scan* for an item. This involves looking through *every item in a table*, so it is less efficient and can take significant time for larger tables.

30. Choose **Scan**, expand **Filters**, then use this filter:

- **Attribute name:**
- **Type**
- **Condition**
- **Value**

31. Choose **Run**.

Only the song released in 1971 is displayed.

# Task 5: Delete the Table

In this task, you will delete the *Music* table, which will also delete all the data in the table.

32. In the left navigation pane, choose **Tables**.
33. Choose the **Music** table.
34. Choose **Actions** , then **Delete table**.
35. Enter 'delete' in the confirmation text box.
36. Choose **Delete table** .

The table will be deleted.

## Conclusion

Congratulations! You have now successfully learned how to:

- Create an Amazon DynamoDB table
- Enter data into an Amazon DynamoDB table
- Query an Amazon DynamoDB table
- Delete an Amazon DynamoDB table