



Load Data into a DynamoDB Table

DI

Dineshraj Dhanapathy

The screenshot shows the AWS DynamoDB console with the 'ContentCatalog' table selected. The table has 6 items returned. The first item is selected, showing details like Id (Number), Authors, ContentType, Difficulty, Price, ProjectCategory, Published, Services, and Title.

Id (Number)	Authors	ContentType	Difficulty	Price	ProjectCategory	Published	Services	Title
3	[{"S": "Ne..."}]	Project	Easy peasy	0	AI/ML	true	Build a Cha...	
2	[{"S": "Ne..."}]	Project	Easy peasy	0	Analytics	true	Visualize da...	
203		Video	0		[{"S": "Am..."}]		AWS x Data...	
202		Video	0				Don't miss ...	
201		Video	0		[{"S": "Am..."}]		AWS Relati...	
1	[{"S": "DD..."}]	Project	Easy peasy	0	Storage	true	Host a Web...	

Introducing Today's Project!

What is Amazon DynamoDB?

Amazon DynamoDB is a fully managed NoSQL database that offers fast performance, scalability, and flexibility. It's useful for handling high-traffic applications, real-time data, and dynamic schemas.

How I used Amazon DynamoDB in this project

In today's project, I used Amazon DynamoDB to create tables, load data, and update records. I also explored batch writing, querying, and managing attributes, gaining hands-on experience with NoSQL databases.

One thing I didn't expect in this project was...

One thing I didn't expect in this project was how flexible DynamoDB is compared to relational databases. Being able to store items with different attributes without a fixed schema was really insightful.

This project took me...

This project took me a few hours to complete, including creating tables, loading data, and updating records. Exploring DynamoDB's flexibility and speed made the learning process engaging and valuable.

Create a DynamoDB table

DynamoDB tables organize data using items and attributes instead of fixed rows and columns. Each item can have a unique set of attributes, offering flexibility not possible in relational databases.

An attribute is a piece of data describing an item in DynamoDB. Each item can have multiple attributes, and unlike relational databases, items in DynamoDB can have unique attribute sets.

DI

Dineshraj Dhanapathy
NextWork Student

NextWork.org

Items returned (1)		Actions	Create item
<input type="checkbox"/>	StudentName (String)		
<input type="checkbox"/>	Nikko	4	

Read and Write Capacity

Read capacity units (RCUs) and write capacity units (WCUs) are measures of DynamoDB's performance. 1 RCU supports 2 reads/sec, while 1 WCU allows 1 write/sec. Usage impacts cost and scalability.

Amazon DynamoDB's Free Tier covers 25GB storage, 25 RCUs, and 25 WCUs, handling 200M requests/month for free. I turned off auto scaling to avoid unexpected costs and control resource allocation.

DI

Dineshraj Dhanapathy
NextWork Student

NextWork.org

Read/write capacity settings [Info](#)

Capacity mode

On-demand
Simplify billing by paying for the actual reads and writes your application performs.

Provisioned
Manage and optimize your costs by allocating read/write capacity in advance.

Read capacity

[Auto scaling](#) | [Info](#)
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

On
 Off

Provisioned capacity units

1

Write capacity

[Auto scaling](#) | [Info](#)
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

On
 Off

Provisioned capacity units

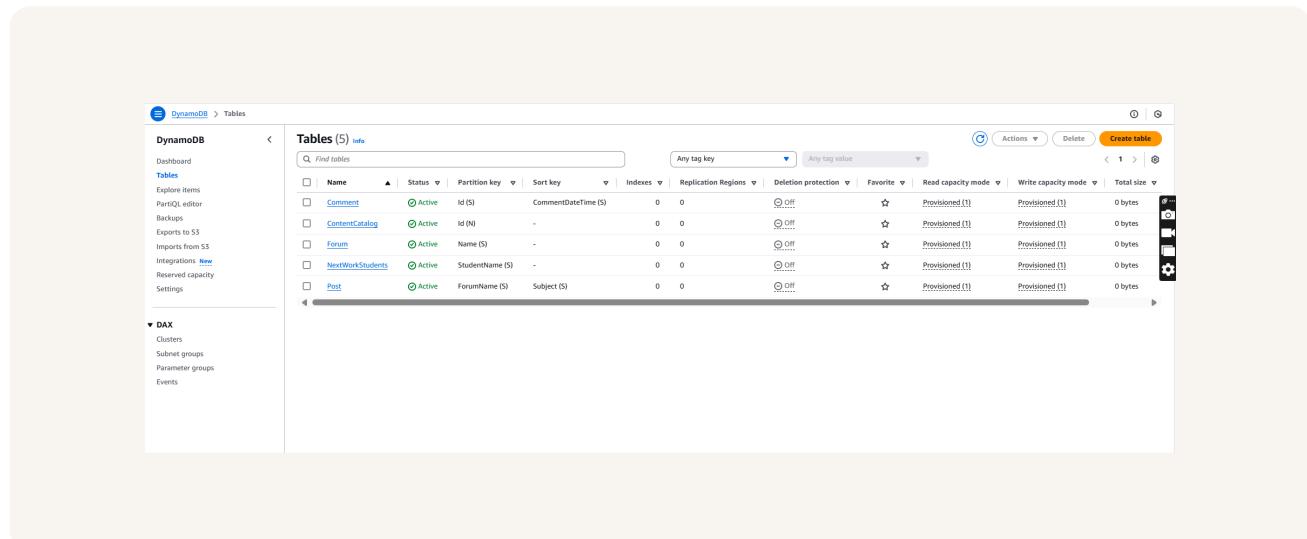
1

Using CLI and CloudShell

AWS CloudShell is a browser-based shell in the AWS Management Console that lets you run commands. It comes with AWS CLI pre-installed, making it easy to manage AWS resources without setup.

AWS CLI is a command-line tool that allows users to create, manage, and automate AWS resources using commands instead of the AWS console, making cloud operations faster and more efficient.

I ran a CLI command in AWS CloudShell that created four DynamoDB tables: ContentCatalog, Forum, Post, and Comment, each with defined attributes, key schemas, and provisioned throughput settings.



Loading Data with CLI

I ran a CLI command in AWS CloudShell that used batch-write-item to load data into four DynamoDB tables: ContentCatalog, Forum, Post, and Comment, by reading data from JSON files.

```
nextworksampled $ aws dynamodb batch-write-item --request-items file://ContentCatalog.json
{
  "UnprocessedItems": {}
}
nextworksampled $ aws dynamodb batch-write-item --request-items file://Forum.json
{
  "UnprocessedItems": {}
}
nextworksampled $ aws dynamodb batch-write-item --request-items file://Post.json
{
  "UnprocessedItems": {}
}
nextworksampled $ aws dynamodb batch-write-item --request-items file://Comment.json
{
  "UnprocessedItems": {}
}
nextworksampled $ █
```

Observing Item Attributes

Attributes		
Attribute name	Value	Type
Id - Partition key	1	Number
Authors	<input type="button" value="Insert a field ▾"/>	List <input type="button" value="Remove"/>
ContentType	Project	String <input type="button" value="Remove"/>
Difficulty	Easy peasy	String <input type="button" value="Remove"/>
Price	0	Number <input type="button" value="Remove"/>
ProjectCategory	Storage	String <input type="button" value="Remove"/>
Published	<input checked="" type="radio"/> True <input type="radio"/> False	Boolean <input type="button" value="Remove"/>
Title	Host a Website on Amazon S3	String <input type="button" value="Remove"/>
URL	aws-host-a-website-on-s3	String <input type="button" value="Remove"/>

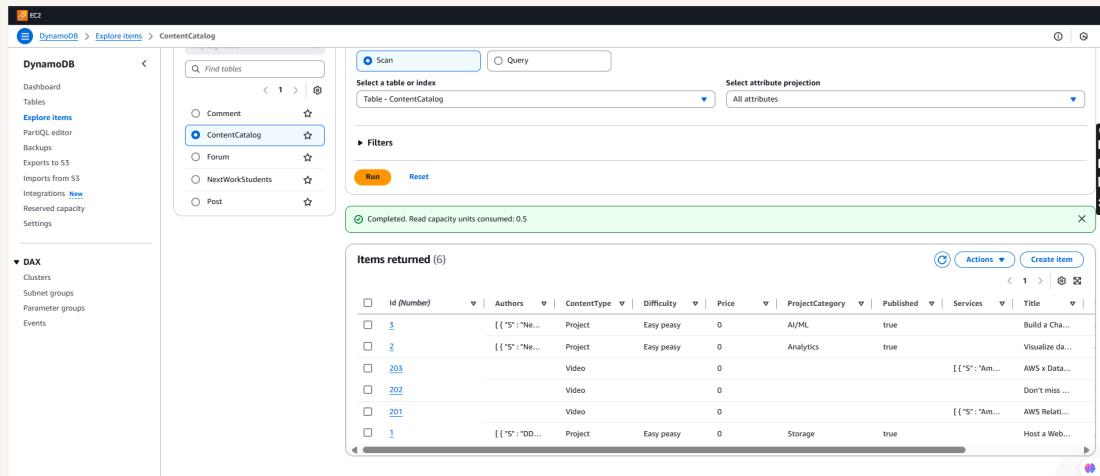
I checked a ContentCatalog item, which had the following attributes: id, content type, difficulty, price, project category, URL, published status, and title—all defining the project's details.

I checked another ContentCatalog item, which had a different set of attributes: Services, title, URL, videotype, etc.,

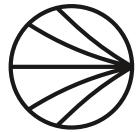
Benefits of DynamoDB

A benefit of DynamoDB over relational databases is flexibility because each item can have unique attributes, unlike relational databases that require uniform columns. This makes it ideal for e-commerce and dynamic data.

Another benefit over relational databases is speed because DynamoDB uses partition keys to quickly locate items, while relational databases must scan entire tables, which slows down performance for large datasets.



Id (Number)	Authors	ContentType	Difficulty	Price	ProjectCategory	Published	Services	Title
3	{ "S": "NextWorkStudent" }	Project	Easy peasy	0	AI/ML	true		Build a Chatbot
2	{ "S": "NextWorkStudent" }	Project	Easy peasy	0	Analytics	true		Visualize data
203		Video		0				AWS Data...
202		Video		0				Don't miss...
201		Video		0				AWS Relati...
1	{ "S": "DD..." }	Project	Easy peasy	0	Storage	true		Host a Web...



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

