



# Load and Query DynamoDB Tables



Prerna Kulkarni

Items returned (6)						
	Id (Number)	Authors	ContentType	Difficulty	Price	ProjectCategory
<input type="checkbox"/>	<a href="#">3</a>	[{"\$": "Ne..."]	Project	Easy peasy	0	AI/ML
<input type="checkbox"/>	<a href="#">2</a>	[{"\$": "Ne..."]	Project	Easy peasy	0	Analytics
<input type="checkbox"/>	<a href="#">203</a>		Video		0	
<input type="checkbox"/>	<a href="#">202</a>		Video		0	
<input type="checkbox"/>	<a href="#">201</a>		Video		0	
<input type="checkbox"/>	<a href="#">1</a>	[{"\$": "Nat..."]	Project	Easy peasy	0	Storage



# Introducing Today's Project!

## What is Amazon DynamoDB?

Amazon DynamoDB is a serverless, NoSQL database service that allows you to develop modern applications at any scale.

## How I used Amazon DynamoDB in this project

By creating tables, using Amazon CLI.

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

Going this far!

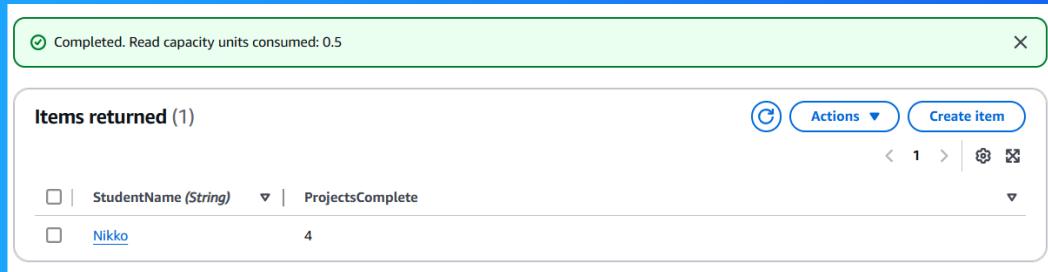
## This project took me...

50 mins.

# Create a DynamoDB table

Each item in a DynamoDB table requires that you create a primary key for the table, as described in the DynamoDB documentation. A primary key can be a partition key or a combination of a partition key and sort key. The primary key must be unique.

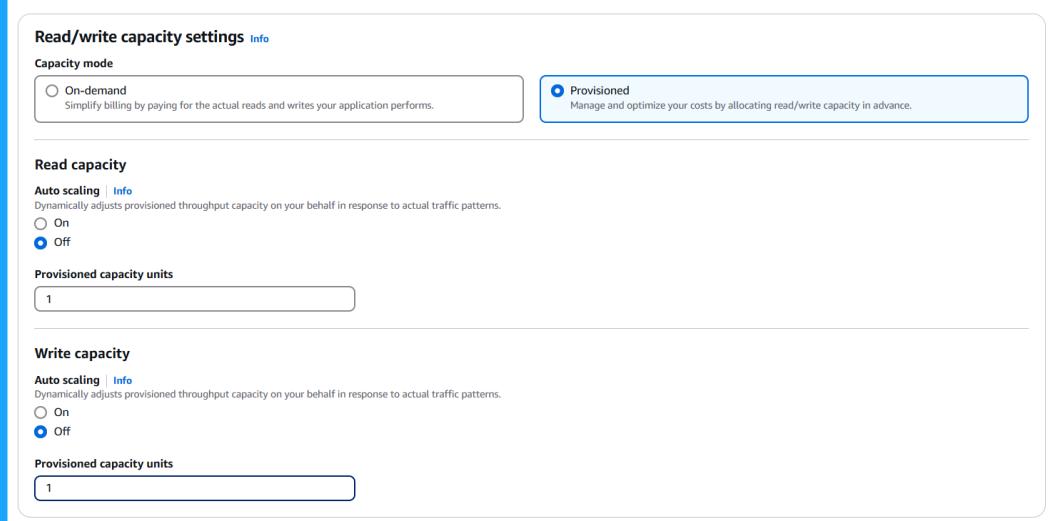
An attribute is like a piece of data about an item



# Read and Write Capacity

Read capacity units (RCUs): to measure how many engines DynamoDB is using to operate. Write capacity units (WCUs) are just like read capacity units - they give your DynamoDB tables the engines to edit/update/delete data

The Free Tier for DynamoDB gives you 25GB of data storage, plus 25 Write and 25 Read Capacity Units (WCU, RCU). This is enough to handle 200M requests per month all for free. To lower the costs I turned off auto scaling.

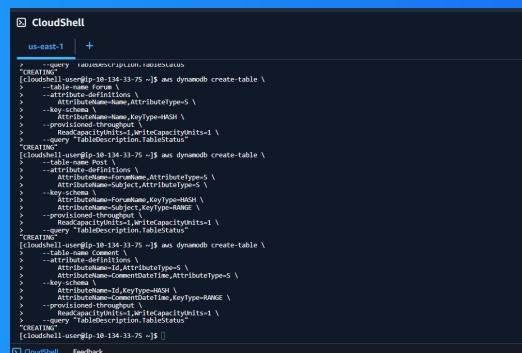


# Using CLI and CloudShell

AWS CLI (Command Line Interface) is a software that lets you create, delete and update AWS resources with commands instead of clicking through your console.

Engineers use the CLI to automate tasks and manage AWS resources efficiently using scripts, making it essential for managing your cloud environment in an efficient way.

Creates four new tables in AWS DynamoDB, each with specific attributes and settings.



```
us-east-1 +  
$ aws dynamodb create-table \  
[cloudshell-user@ip-10-134-33-75 ~]$ aws dynamodb create-table \  
  >   --attribute-definitions \  
  >     AttributeDefinition<br>AttributeType=S \  
  >     AttributeDefinition<br>AttributeType=S \  
  >   .key-schema ValueName=Id,KeyType=HASH \  
  >   .provisioned-throughput ValueName=ReadCapacityUnits,Value=1,ValueName=WriteCapacityUnits,Value=1 \  
  >   .query TableDescription.tableStatus=1 \  
'CREATING'  
[cloudshell-user@ip-10-134-33-75 ~]$ aws dynamodb create-table \  
  >   --attribute-definitions \  
  >     AttributeDefinition<br>AttributeType=S \  
  >     AttributeDefinition<br>AttributeType=S \  
  >   .key-schema ValueName=Comments,KeyType=HASH \  
  >   .attribute-name-alias CommentsAttribute,AttributeType=S \  
  >   .attribute-name-alias Id,AttributeType=S \  
  >   .provisioned-throughput ValueName=ReadCapacityUnits,Value=1,ValueName=WriteCapacityUnits,Value=1 \  
  >   .query TableDescription.tableStatus=1 \  
'CREATING'  
[cloudshell-user@ip-10-134-33-75 ~]$ aws dynamodb create-table \  
  >   --attribute-definitions \  
  >     AttributeDefinition<br>AttributeType=S \  
  >     AttributeDefinition<br>AttributeType=S \  
  >   .key-schema ValueName=Comments,KeyType=HASH \  
  >   .attribute-name-alias CommentsAttribute,AttributeType=S \  
  >   .attribute-name-alias Id,AttributeType=S \  
  >   .provisioned-throughput ValueName=ReadCapacityUnits,Value=1,ValueName=WriteCapacityUnits,Value=1 \  
  >   .query TableDescription.tableStatus=1 \  
'CREATING'  
[cloudshell-user@ip-10-134-33-75 ~]$ |
```



# Loading Data with CLI

```
aws dynamodb batch-write-item --request-items file://ContentCatalog.json
aws dynamodb batch-write-item --request-items file://Forum.json aws
dynamodb batch-write-item --request-items file://Post.json aws dynamodb
batch-write-item --request-items f
```

```
        }
    }
}
[cloudshell-user@ip-10-134-33-75 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://ContentCatalog.json
{
  "UnprocessedItems": {}
}
[cloudshell-user@ip-10-134-33-75 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://Forum.json
{
  "UnprocessedItems": {}
}
[cloudshell-user@ip-10-134-33-75 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://Post.json
{
  "UnprocessedItems": {}
}
[cloudshell-user@ip-10-134-33-75 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://Comment.json
{
  "UnprocessedItems": {}
}
[cloudshell-user@ip-10-134-33-75 nextworksampleddata]$
```

# 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"/>

Projects and Videos.

Price, Services, Title, URL, VideoType

# Benefits of DynamoDB

Relational databases use SQL, which makes handling complex queries a lot more straightforward .The strictness of a relational database's schema also means data is kept precise, accurate and consistent.

Scalability is a core tenet of DynamoDB's design philosophy, with built-in support for auto-scaling throughput capacity based on workload demand.

Items returned (6)						
	Id (Number)	Authors	ContentType	Difficulty	Price	ProjectCategory
<input type="checkbox"/>	<a href="#">3</a>	[{"S": "Ne...}	Project	Easy peasy	0	AI/ML
<input type="checkbox"/>	<a href="#">2</a>	[ {"S": "Ne...	Project	Easy peasy	0	Analytics
<input type="checkbox"/>	<a href="#">203</a>		Video		0	
<input type="checkbox"/>	<a href="#">202</a>		Video		0	
<input type="checkbox"/>	<a href="#">201</a>		Video		0	
<input type="checkbox"/>	<a href="#">1</a>	[ {"S": "Nat...	Project	Easy peasy	0	Storage



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

