

**Repositórios de Dados e NoSQL**  
**eEDB-016/2024-2**  
**Aluno: Marcelo Dozzi Barbugli**

hands-on AWS DynamoDB - parte 1

Criando a tabela 'sells', chaves: 'user(s)' e 'order(n)', configurando de forma personalizada, com capacidade de leitura/escrita de 3 itens por segundo (consistência), capacidade de leitura.escrita será de min 1 unidade e max 2 unidades. O index 'foo-index' é do tipo local, sort key = 'foo'(s). O index global é o 'order-user-index', partition key = 'order' (n), sort key = 'user' (s).

## Create table

### Table details [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

#### Table name

This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.).

#### Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

1 to 255 characters and case sensitive.

#### Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

1 to 255 characters and case sensitive.

### Table settings

☐ Default settings

The fastest way to create your table. You can modify these settings now or after your table has been created.

☒ Customize settings

Use these advanced features to make DynamoDB work better for your needs.

## ▼ Capacity calculator

Average item size (KB)

Item read/second

Read consistency

Eventually consistent ▼

Item write/second

Write consistency

Standard ▼

Read capacity units

2

Write capacity units

3

Region

us-east-1

Estimated cost

\$1.65 / month

## Read/write capacity settings [Info](#)

Capacity mode



Provisioned

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



On-demand

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

### Read capacity

Auto scaling [Info](#)

Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

☒ On

☐ Off

Minimum capacity units

Maximum capacity units

Target utilization (%)

### Write capacity

Auto scaling [Info](#)

Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

☒ On

☐ Off

Minimum capacity units

Maximum capacity units

Target utilization (%)

Secondary indexes [Info](#)

Delete

Create local index

Create global index

<input type="checkbox"/>	Name	Type	Partition key	Sort key	Projected attributes
<input type="checkbox"/>	<a href="#">foo-index</a>	Local	-	foo (String)	All
<input type="checkbox"/>	<a href="#">order-user-index</a>	Global	order (Number)	user (String)	All

Encryption at rest [Info](#)

All user data stored in Amazon DynamoDB is fully encrypted at rest. By default, Amazon DynamoDB manages the encryption key, and you are not charged any fee for using it.

Encryption key management

☐ Owned by Amazon DynamoDB [Learn more](#)

The AWS KMS key is owned and managed by DynamoDB. You are not charged an additional fee for using this key.

☒ AWS managed key [Learn more](#)

Key alias: aws/dynamodb. The key is stored in your account and is managed by AWS Key Management Service (AWS KMS). AWS KMS charges apply.

☐ Stored in your account, and owned and managed by you [Learn more](#)

The key is stored in your account and is owned and managed by you. AWS KMS charges apply.

Apos a criação da tabela, iremos adiciona um item.

Create item 

FormJSON view

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. [Learn more](#)

Attributes

Add new attribute

Attribute name	Value	Type
user - Partition key	<input type="text" value="mengano"/>	String
order - Sort key	<input type="text" value="1009"/>	Number

Cancel

Create item

Realizando um Scan em todos os itens da tabela 'sells'

▼ Scan or query items

Scan

Query

Select a table or index

Table - sells

Select attribute projection

All attributes

► Filters

Run

Reset

Completed. Read capacity units consumed: 0.5

Items returned (1)

Refresh

Actions

Create item

<

1

>

⌕

⌵

☐

user (String)

▼

order (Number)

▼

☐

mengano

1009

realizando um Scan com filtro:

▼ Scan or query items

Scan

Query

Select a table or index

Table - sells

Select attribute projection

All attributes

▼ Filters

Attribute name

Q user

Type

String

Condition

Equal to

Value

mengano

Remove

Add filter

Run

Reset

Completed. Read capacity units consumed: 0.5

Items returned (1)

Refresh

Actions

Create item

<

1

>

⌕

⌵

☐

user (String)

▼

order (Number)

▼

☐

mengano

1009

Realizando uma Query:

▼ Scan or query items

Scan

Query

Select a table or index

Table - sells

Select attribute projection

All attributes

user (Partition key)

mengano

order (Sort key)

Equal to

Enter sort key value

☐ Sort descending

► Filters

Run

Reset

Completed. Read capacity units consumed: 0.5

Items returned (1)

☐

user (String)

▼

order (Number)

▼

☐

mengano

1009

## Adicionando um novo atributo na tabela 'sells'

Edit item

FormJSON view

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. [Learn more](#)

Attributes

Add new attribute

Attribute name	Value	Type	
user - Partition key	<input type="text" value="mengano"/>	String	
order - Sort key	<input type="text" value="1009"/>	Number	
<input type="text" value="age"/>	<input type="text" value="19"/>	Number	<div>Remove</div>

Cancel

Save

Save and close

## realizando uma query na tabela 'sells' utilizando filtro:

Scan

Query

Select a table or index

Table - sells

Select attribute projection

All attributes

user (Partition key)

order (Sort key)

Equal to

Enter sort key value

☐ Sort descending

Filters

Attribute name

age

Type

Number

Condition

Equal to

Value

19

Remove

Add filter

Run

Reset

Completed. Read capacity units consumed: 0.5

Items returned (1)

Actions

Create item

<1>

	user (String)	order (Number)	age
<input type="checkbox"/>	<a href="#">mengano</a>	1009	19

## Realizando query no index 'order-user-index':

Scan

Query

Select a table or index

Index - order-user-index

Select attribute projection

Projected attributes

order (Partition key)

user (Sort key)

Equal to

Enter sort key value

☐ Sort descending

Filters

Add a filter to get started.

Add filter

Run

Reset

Completed. Read capacity units consumed: 0.5

Items returned (2)

Actions

	user (String)	order (Number)	age
<input type="checkbox"/>	<a href="#">jose</a>	1000	
<input type="checkbox"/>	<a href="#">maria</a>	1000	55

Usando SQL:  
Scan na tabela 'sells':

Tables (2)

Find tables

< 1 > ⚙

▶ Books

...

▶ sells

...

Query 1 X

✓

Query 2 X

+

1 SELECT \* FROM "sells"

Run

Clear

Table view

JSON view

✓ Completed

Started on 6/3/2024, 6:43:48 PM

Elapsed time 192ms

Items returned (3)

Find items

user	order	age
jose	1000	
maria	1000	55