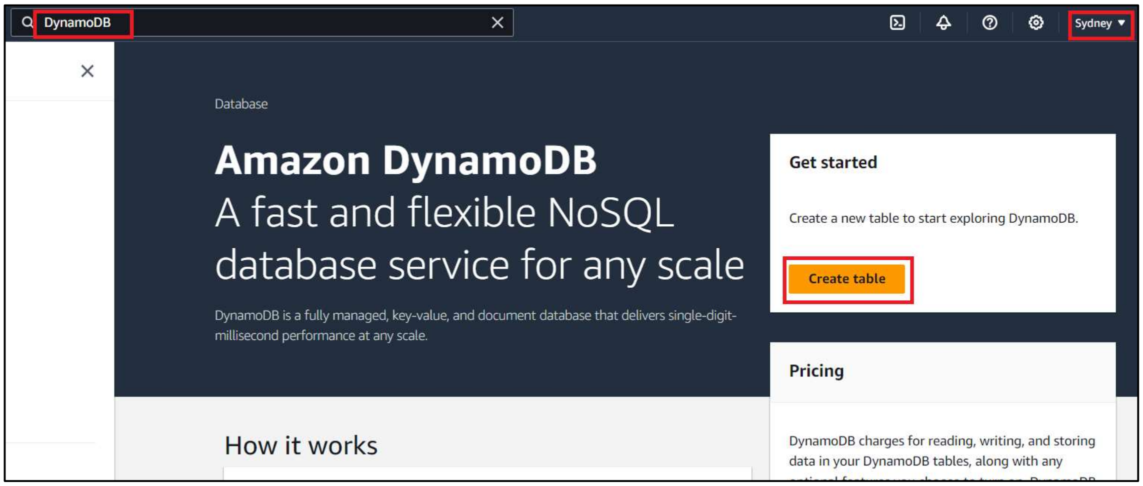
1) Create a DynamoDB Table



Amazon DynamoDB
A fast and flexible NoSQL database service for any scale

DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.

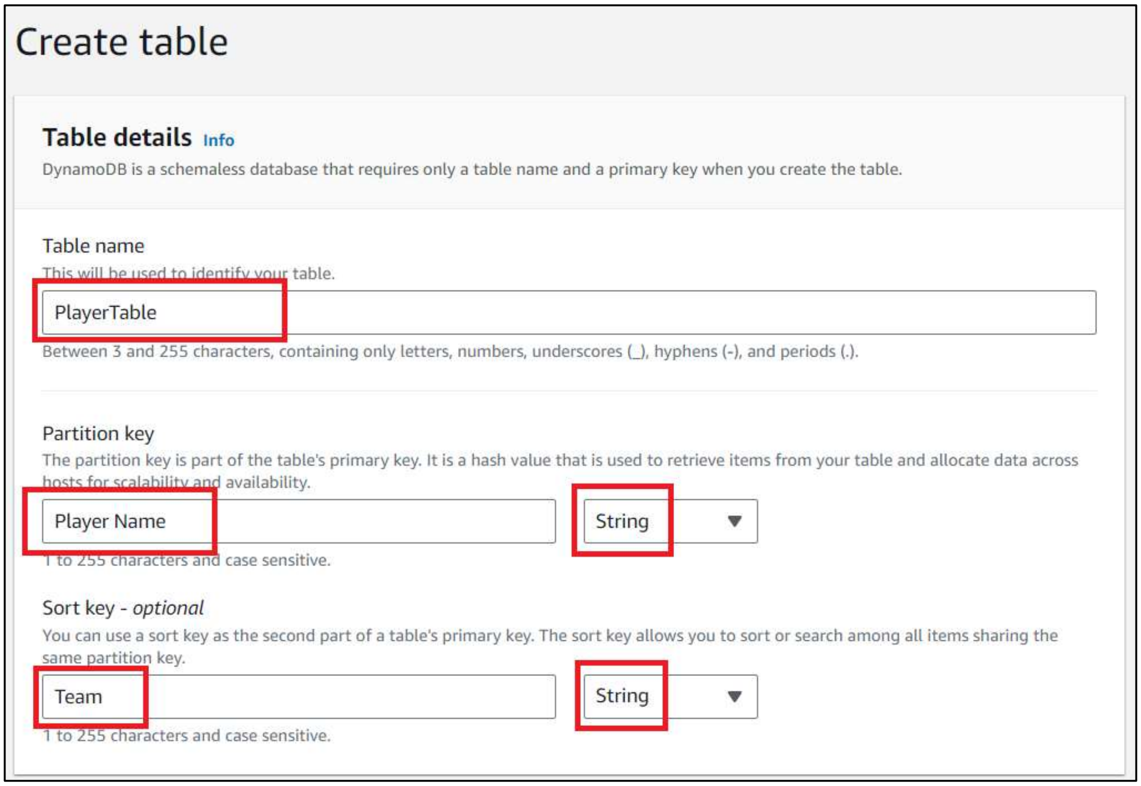
Get started

Create a new table to start exploring DynamoDB.

Pricing

DynamoDB charges for reading, writing, and storing data in your DynamoDB tables, along with any

**Create a players' table which is related to soccer, and is for an application which gives players' rating:**



Create table

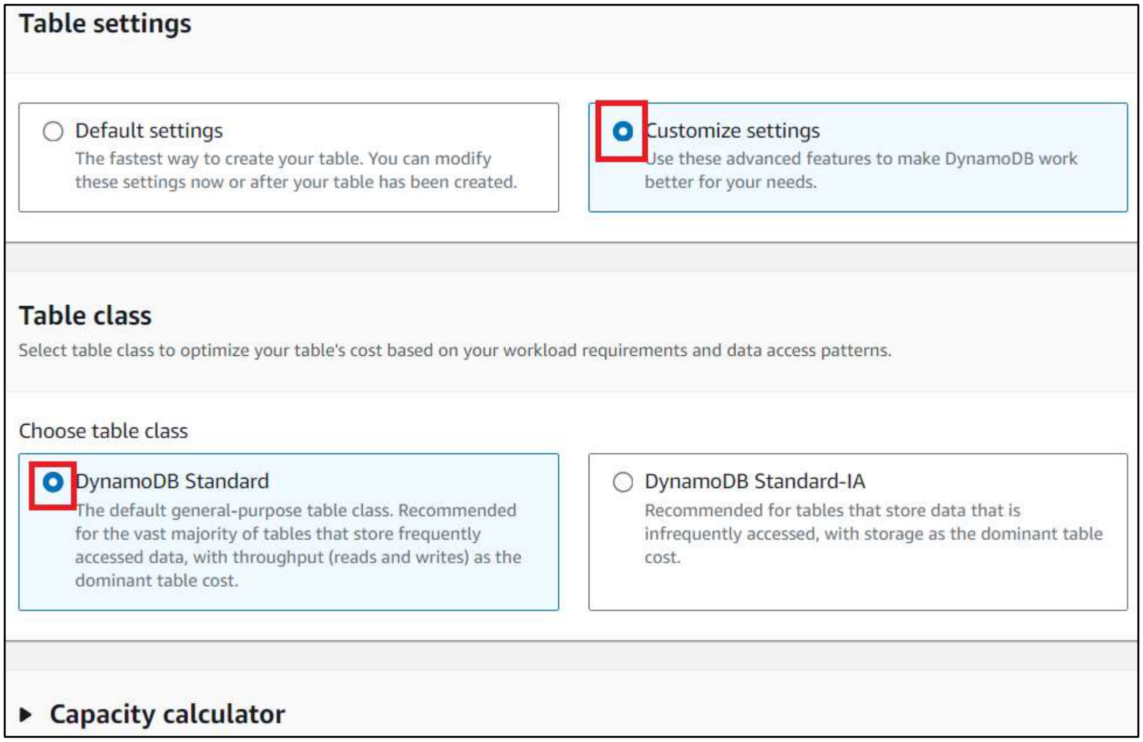
Table details Info

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.
PlayerTable
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.
Player Name — String
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.
Team — String
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.

## Table class

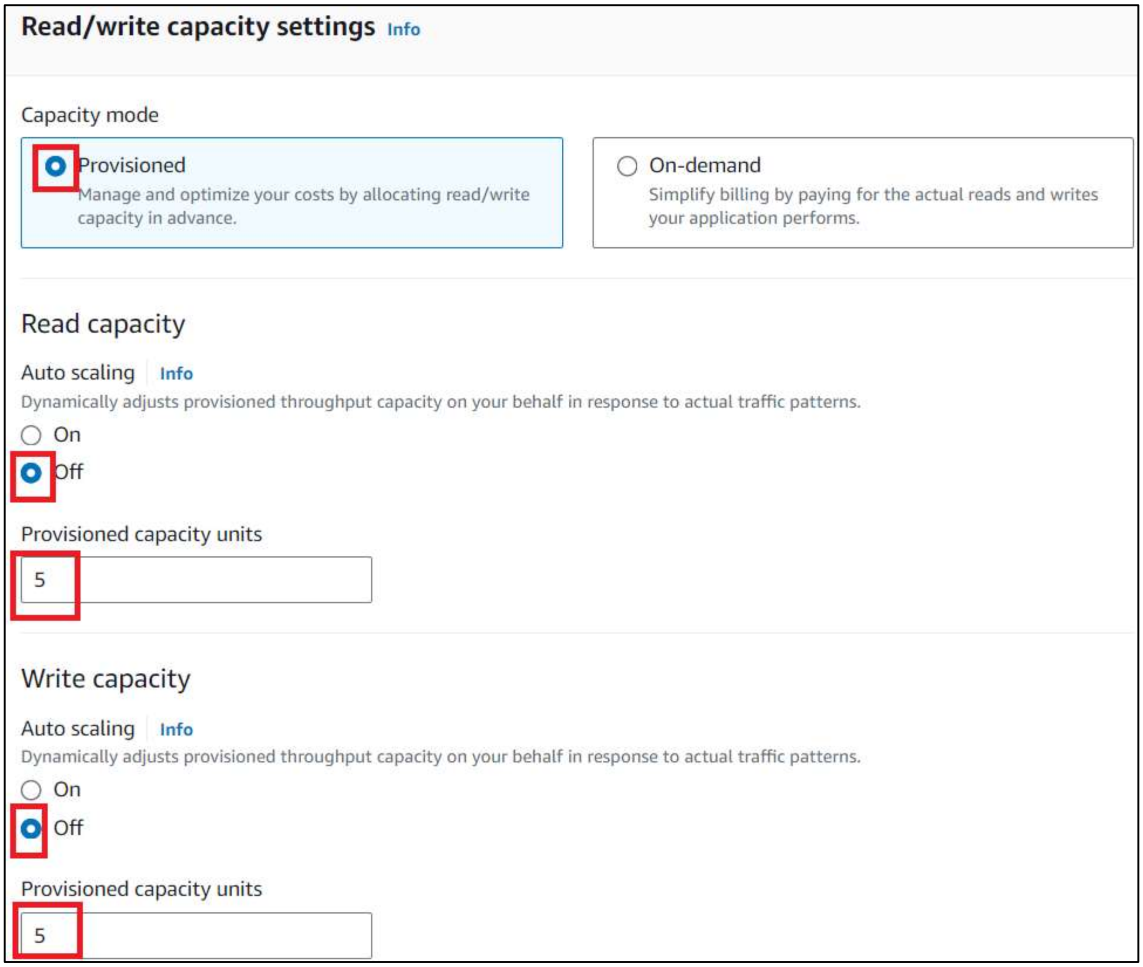
Select table class to optimize your table's cost based on your workload requirements and data access patterns.

Choose table class

● DynamoDB Standard
The default general-purpose table class. Recommended for the vast majority of tables that store frequently accessed data, with throughput (reads and writes) as the dominant table cost.

○ DynamoDB Standard-IA
Recommended for tables that store data that is infrequently accessed, with storage as the dominant table cost.

▶ Capacity calculator



## Read/write capacity settings

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

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

○ On
● Off

Provisioned capacity units: 5

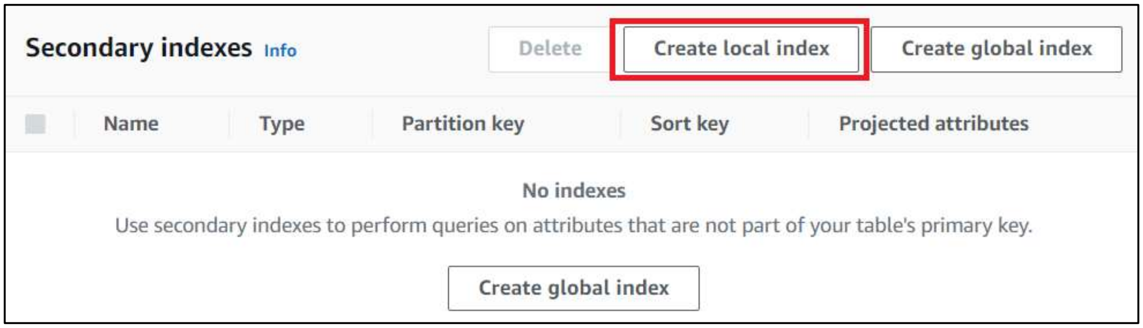
### Write capacity

Auto scaling

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

○ On
● Off

Provisioned capacity units: 5

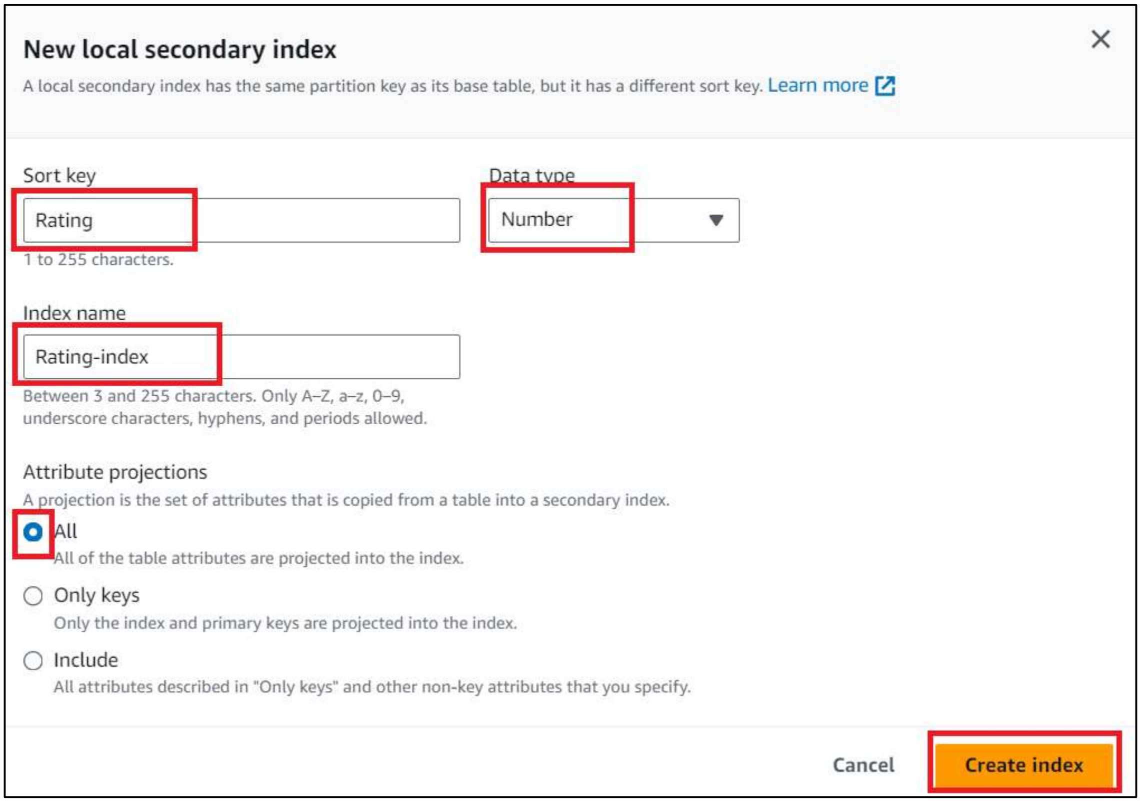


Secondary indexes Info

| ☐ | Name | Type | Partition key | Sort key | Projected attributes |
|---|---|---|---|---|---|

No indexes

Use secondary indexes to perform queries on attributes that are not part of your table's primary key.


New local secondary index

A local secondary index has the same partition key as its base table, but it has a different sort key. Learn more

Sort key: Rating

1 to 255 characters.

Data type: Number

Index name: Rating-index

Between 3 and 255 characters. Only A–Z, a–z, 0–9, underscore characters, hyphens, and periods allowed.

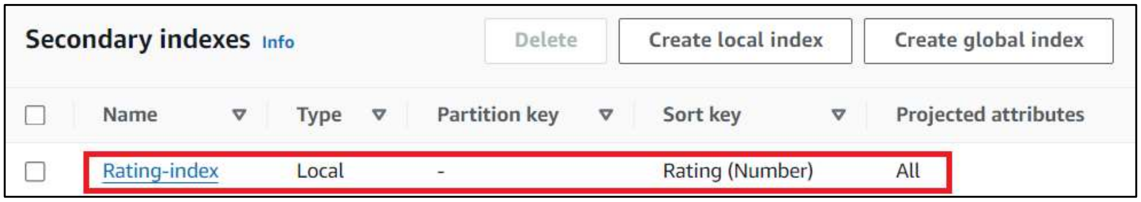
Attribute projections

A projection is the set of attributes that is copied from a table into a secondary index.

● All
All of the table attributes are projected into the index.

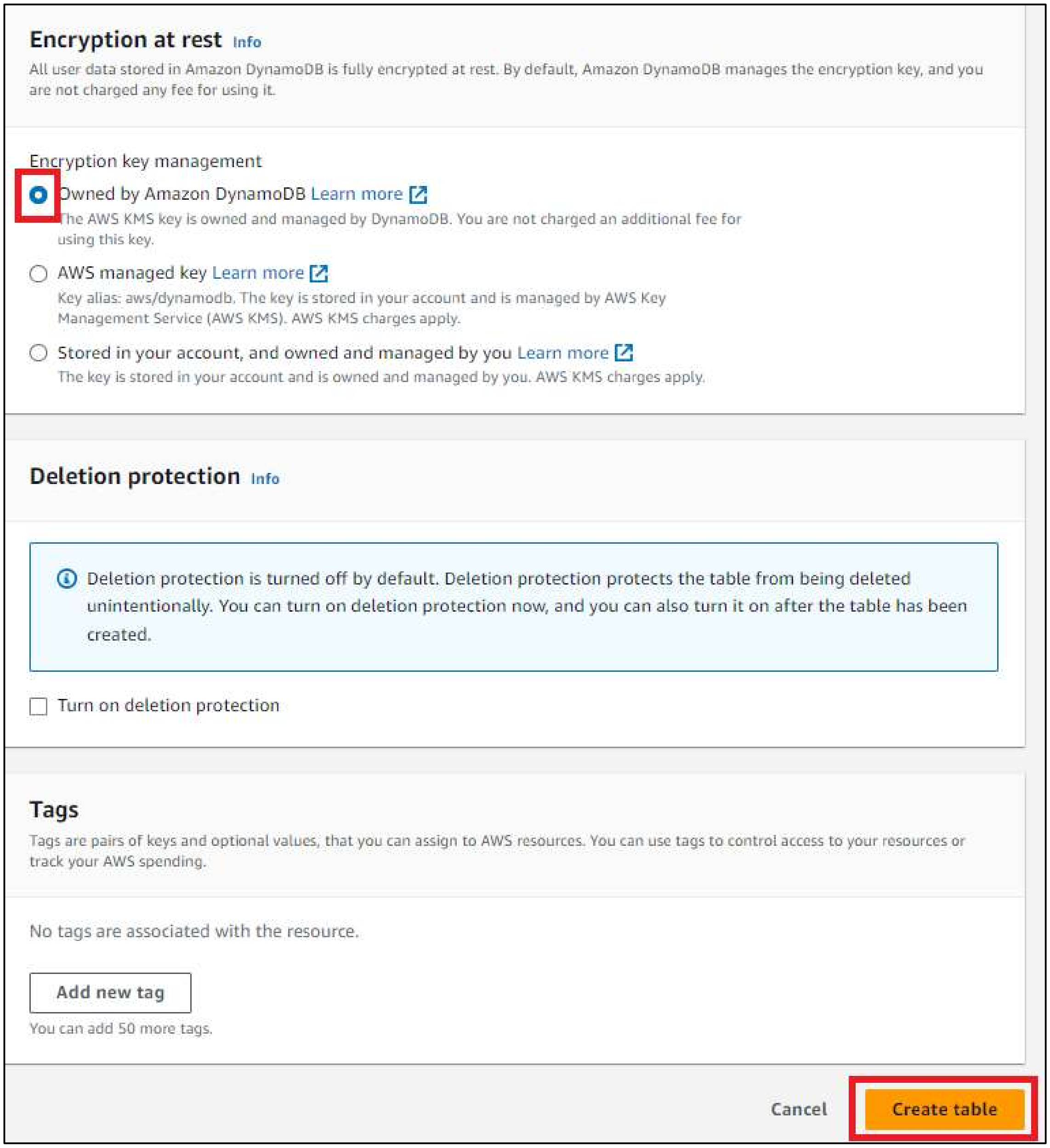
○ Only keys
Only the index and primary keys are projected into the index.

○ Include
All attributes described in "Only keys" and other non-key attributes that you specify.



Secondary indexes Info

| ☐ | Name | Type | Partition key | Sort key | Projected attributes |
|---|---|---|---|---|---|
| ☐ | Rating-index | Local | - | Rating (Number) | 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.

### Deletion protection Info

Deletion protection is turned off by default. Deletion protection protects the table from being deleted unintentionally. You can turn on deletion protection now, and you can also turn it on after the table has been created.

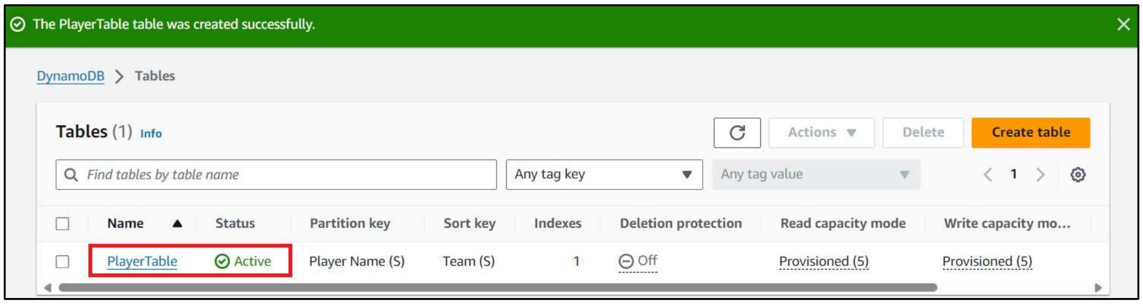
□ Turn on deletion protection

### Tags

Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

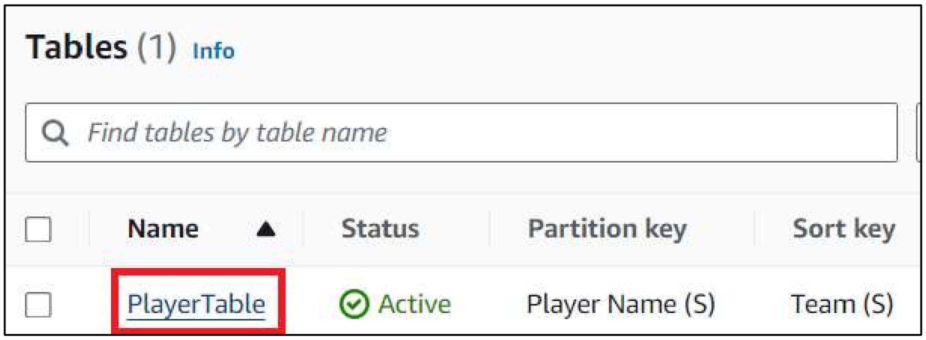
You can add 50 more tags.



The PlayerTable table was created successfully.

| □ | Name ▲ | Status | Partition key | Sort key | Indexes | Deletion protection | Read capacity mode | Write capacity mo... |
|---|---|---|---|---|---|---|---|---|
| □ | PlayerTable | Active | Player Name (S) | Team (S) | 1 | Off | Provisioned (5) | Provisioned (5) |

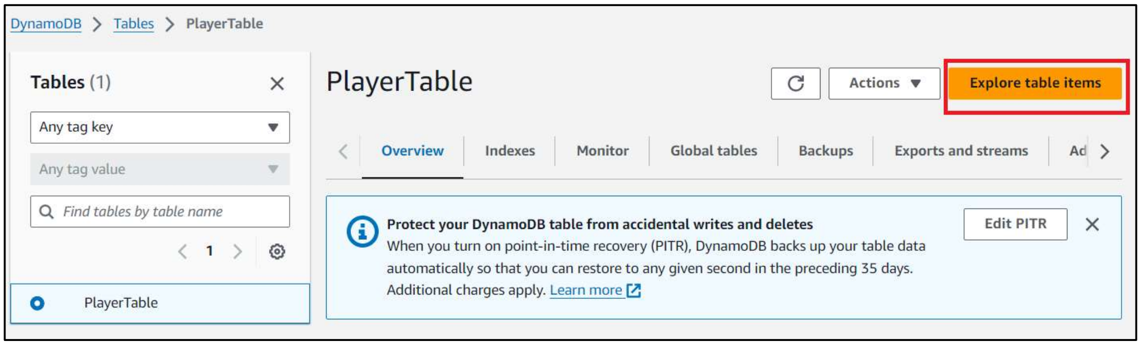
2) Write data to the table



Tables (1) Info

Find tables by table name

| □ | Name ▲ | Status | Partition key | Sort key |
|---|---|---|---|---|
| □ | PlayerTable | ✓ Active | Player Name (S) | Team (S) |



DynamoDB > Tables > PlayerTable

Tables (1)

Any tag key

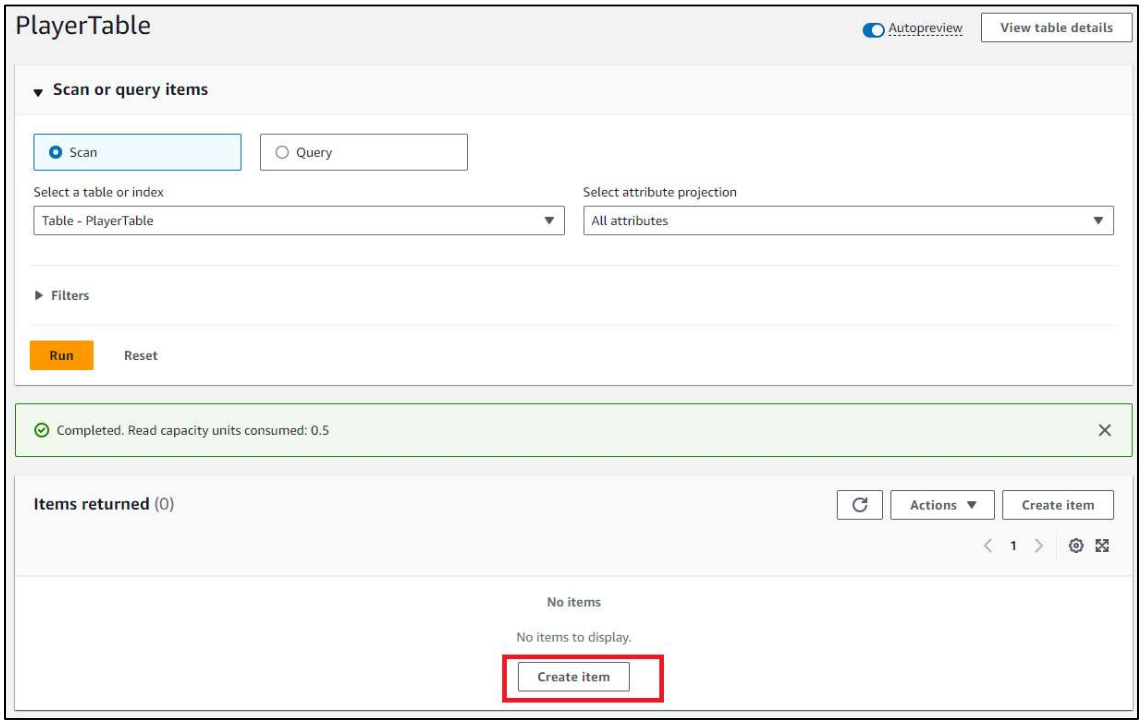
Any tag value

Find tables by table name

○ PlayerTable

PlayerTable

**Protect your DynamoDB table from accidental writes and deletes**
When you turn on point-in-time recovery (PITR), DynamoDB backs up your table data automatically so that you can restore to any given second in the preceding 35 days. Additional charges apply. Learn more



PlayerTable

Scan or query items

● Scan ○ Query

Select a table or index: Table - PlayerTable

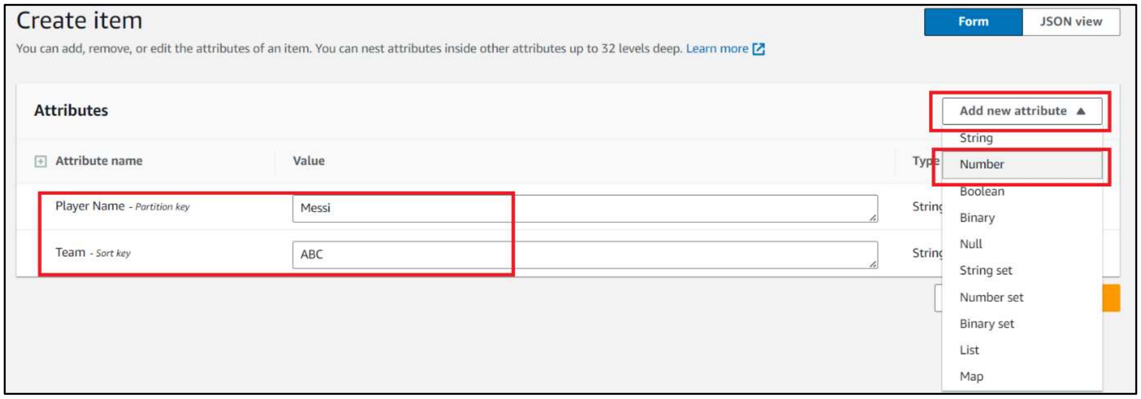
Select attribute projection: All attributes

Completed. Read capacity units consumed: 0.5

Items returned (0)

No items

No items to display.

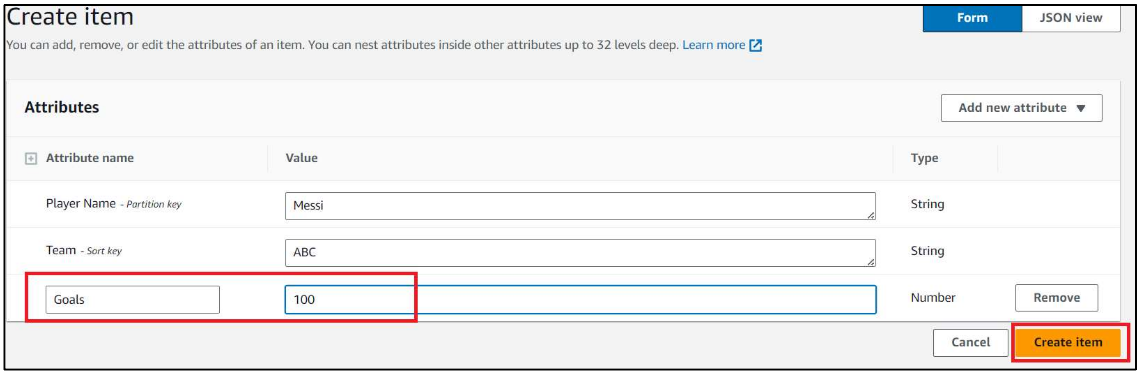


Create item

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

| Attribute name | Value | Type |
|---|---|---|
| Player Name - *Partition key* | Messi | String |
| Team - *Sort key* | ABC | String |

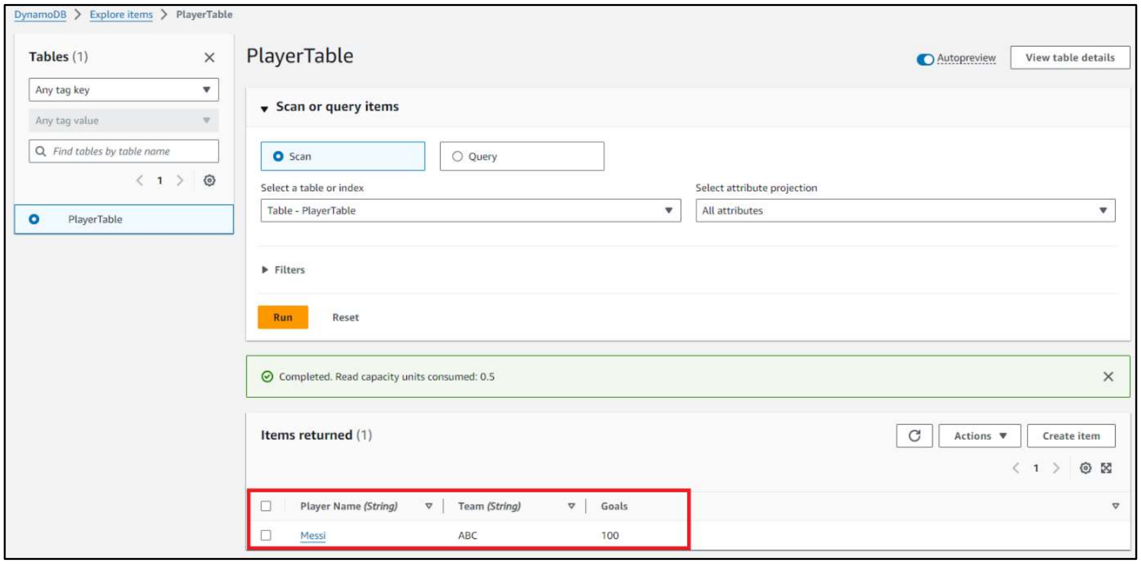


Create item

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

| Attribute name | Value | Type | |
|---|---|---|---|
| Player Name - *Partition key* | Messi | String | |
| Team - *Sort key* | ABC | String | |
| Goals | 100 | Number | Remove |



DynamoDB > Explore items > PlayerTable

Tables (1)

○ PlayerTable

PlayerTable

Scan or query items

● Scan ○ Query

Select a table or index: Table - PlayerTable

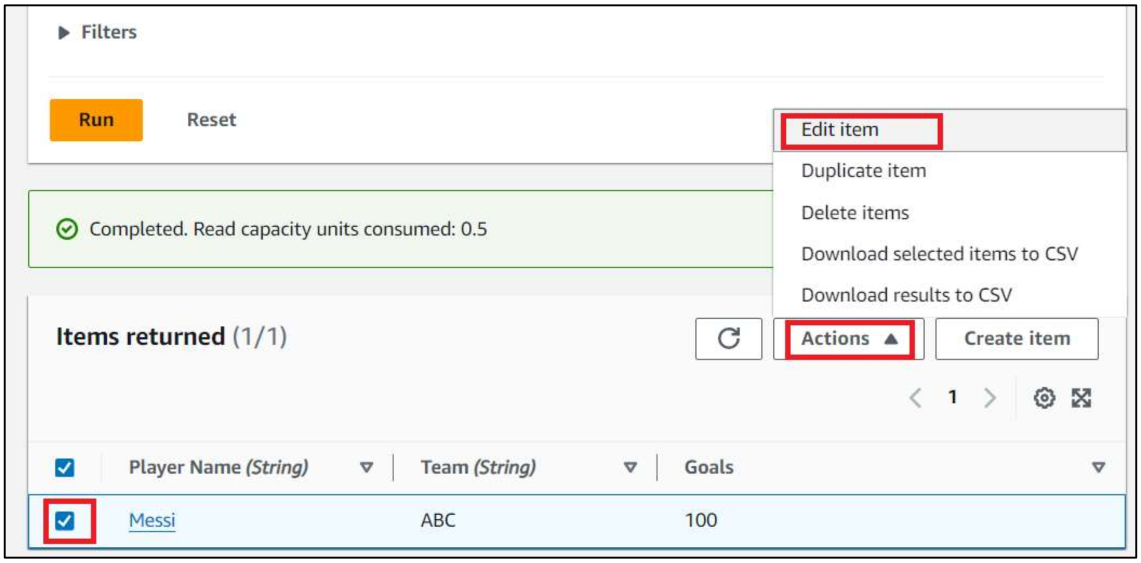
Select attribute projection: All attributes

Completed. Read capacity units consumed: 0.5

Items returned (1)

| □ | Player Name *(String)* | Team *(String)* | Goals |
|---|---|---|---|
| □ | Messi | ABC | 100 |

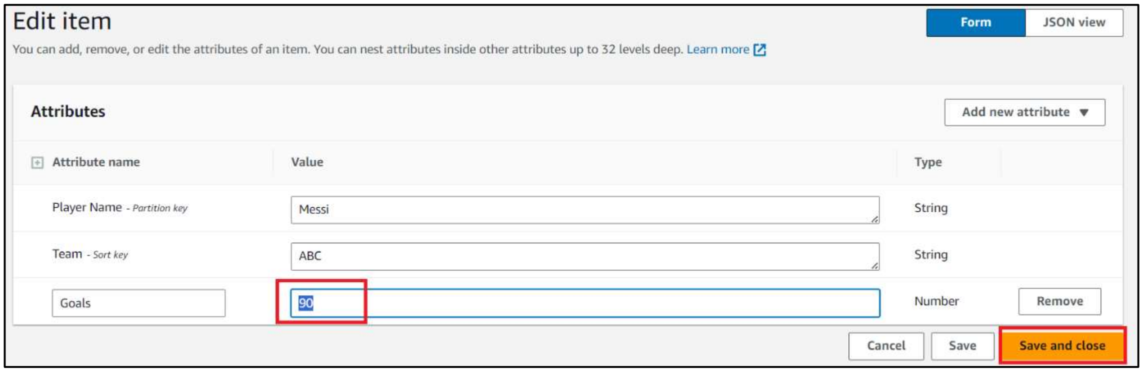
3) Update data in the table



Completed. Read capacity units consumed: 0.5

**Items returned** (1/1)

| ☑ | Player Name *(String)* | Team *(String)* | Goals |
|---|---|---|---|
| ☑ | Messi | ABC | 100 |

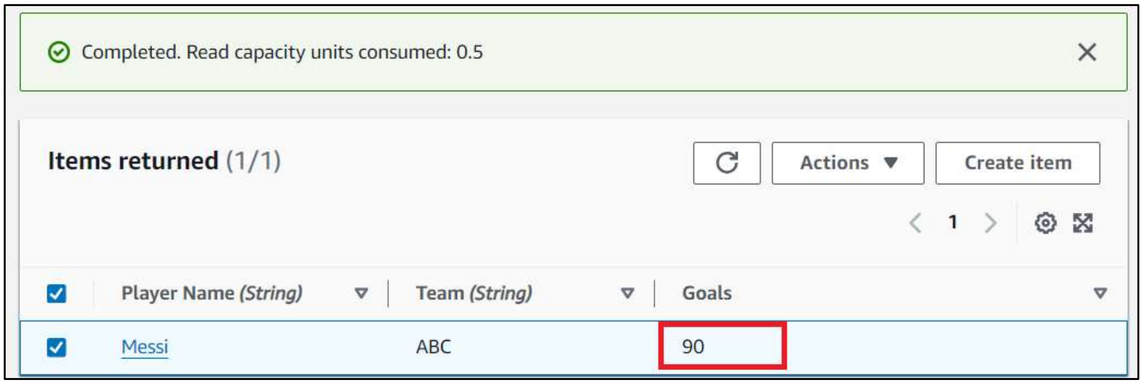


Edit item

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

| Attribute name | Value | Type |
|---|---|---|
| Player Name - *Partition key* | Messi | String |
| Team - *Sort key* | ABC | String |
| Goals | 90 | Number |

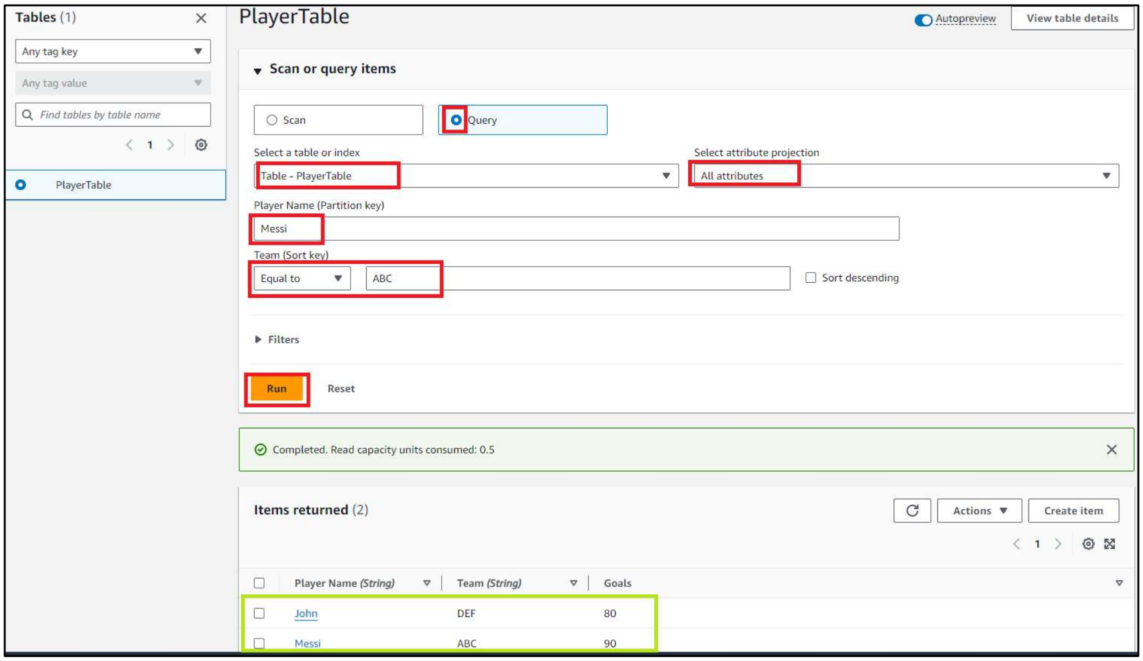


Completed. Read capacity units consumed: 0.5

**Items returned** (1/1)

| ☑ | Player Name *(String)* | Team *(String)* | Goals |
|---|---|---|---|
| ☑ | Messi | ABC | 90 |

4) Query the data in the table



PlayerTable

Scan or query items

○ Scan ● Query

Select a table or index: Table - PlayerTable

Select attribute projection: All attributes

Player Name (Partition key): Messi

Team (Sort key): Equal to ABC ☐ Sort descending

Filters

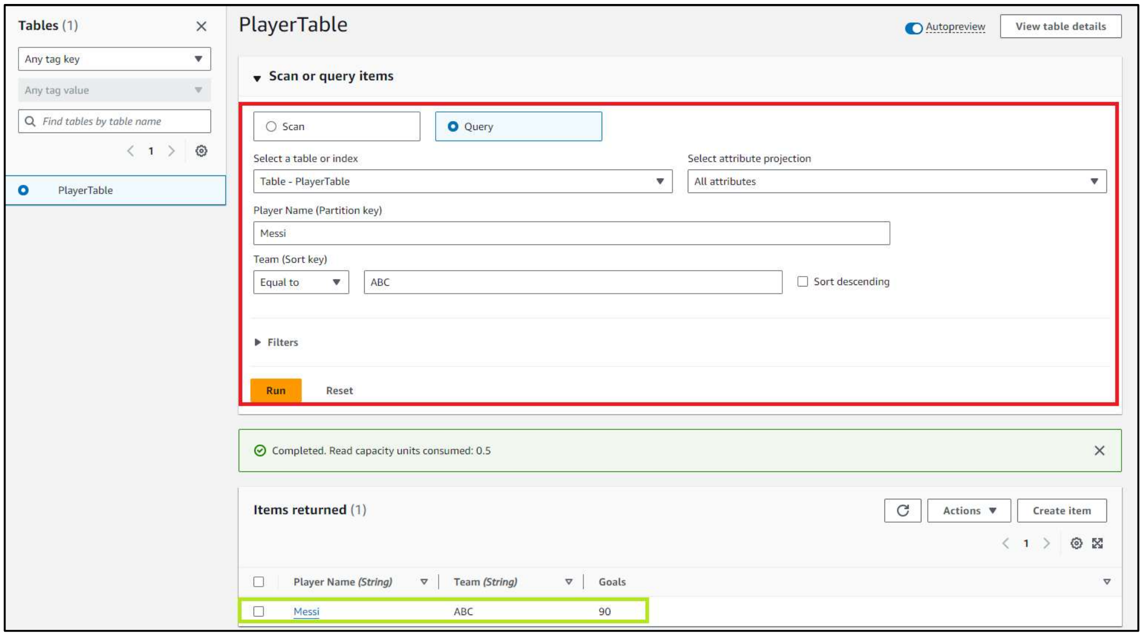
Run

Completed. Read capacity units consumed: 0.5

Items returned (2)

| Player Name (String) | Team (String) | Goals |
|---|---|---|
| John | DEF | 80 |
| Messi | ABC | 90 |

**Query output:**



PlayerTable

Scan or query items

○ Scan ● Query

Select a table or index: Table - PlayerTable

Select attribute projection: All attributes

Player Name (Partition key): Messi

Team (Sort key): Equal to ABC ☐ Sort descending

Filters

Run

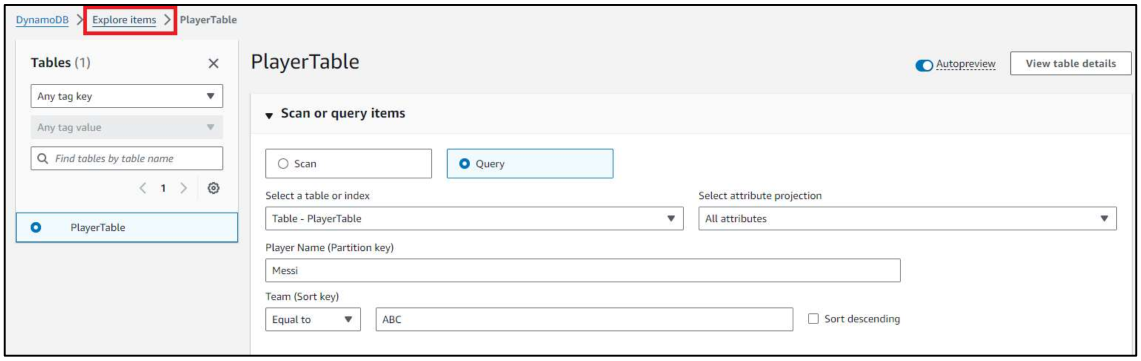
Completed. Read capacity units consumed: 0.5

Items returned (1)

| Player Name (String) | Team (String) | Goals |
|---|---|---|
| Messi | ABC | 90 |

5) Create and Query Global Secondary Index

**Create Global Secondary Index:**



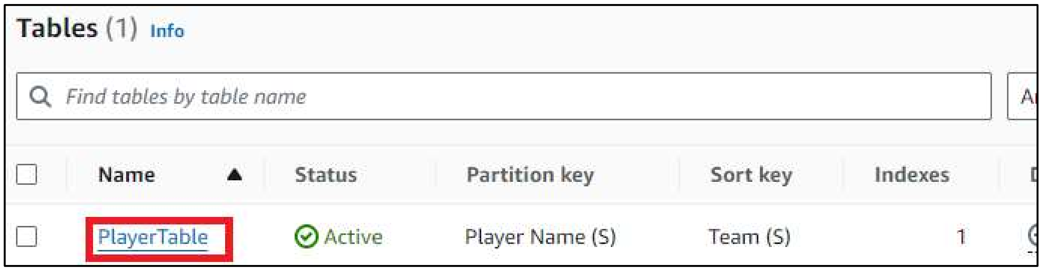
Select a table or index: Table - PlayerTable

Select attribute projection: All attributes

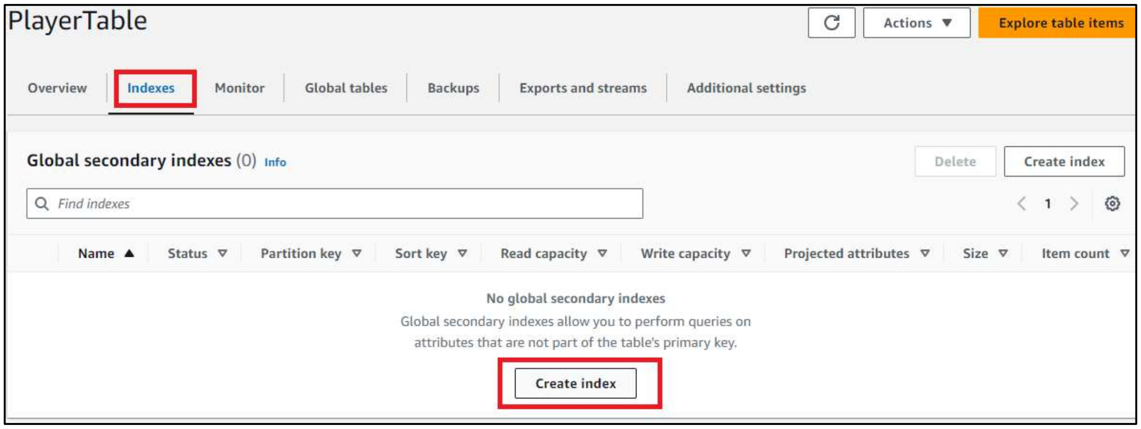
Player Name (Partition key): Messi

Team (Sort key): Equal to ABC

□ Sort descending

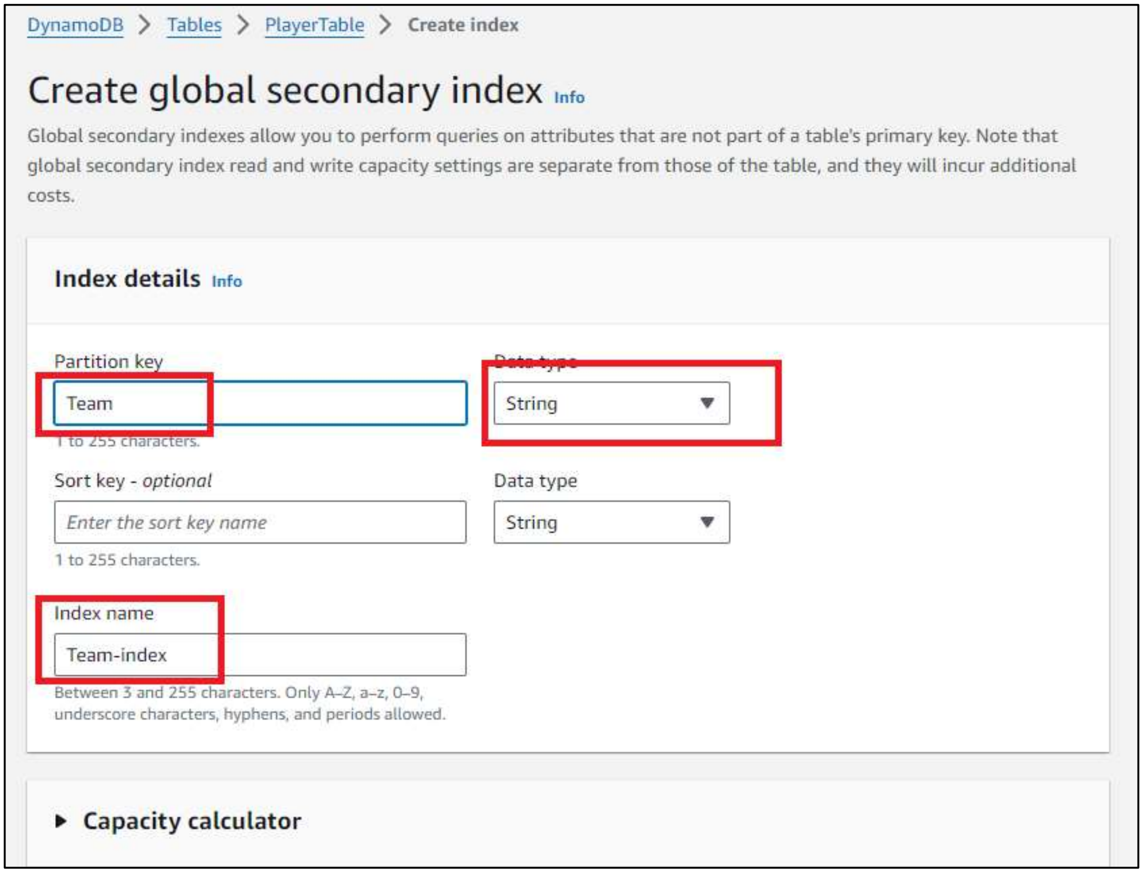


| □ | Name | Status | Partition key | Sort key | Indexes |
|---|---|---|---|---|---|
| □ | PlayerTable | Active | Player Name (S) | Team (S) | 1 |



No global secondary indexes

Global secondary indexes allow you to perform queries on attributes that are not part of the table's primary key.



DynamoDB > Tables > PlayerTable > Create index

Create global secondary index Info

Global secondary indexes allow you to perform queries on attributes that are not part of a table's primary key. Note that global secondary index read and write capacity settings are separate from those of the table, and they will incur additional costs.

Index details Info

Partition key: Team
1 to 255 characters.

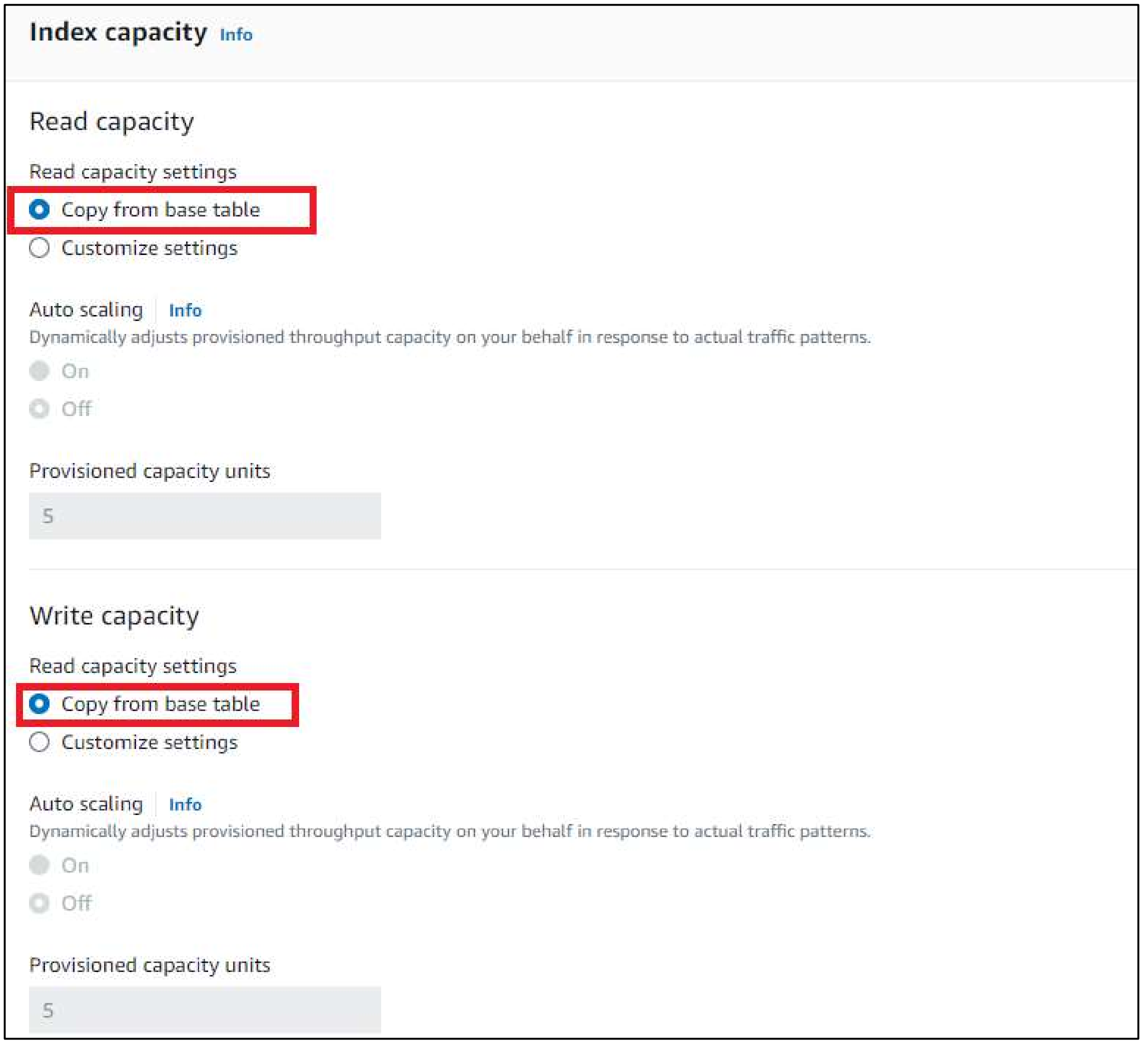
Data type: String

Sort key - *optional*: *Enter the sort key name*
1 to 255 characters.

Data type: String

Index name: Team-index
Between 3 and 255 characters. Only A–Z, a–z, 0–9, underscore characters, hyphens, and periods allowed.

▸ Capacity calculator



### Index capacity Info

Read capacity

Read capacity settings

● Copy from base table

○ Customize settings

Auto scaling | Info

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

○ On

○ Off

Provisioned capacity units

5

Write capacity

Read capacity settings

● Copy from base table

○ Customize settings

Auto scaling | Info

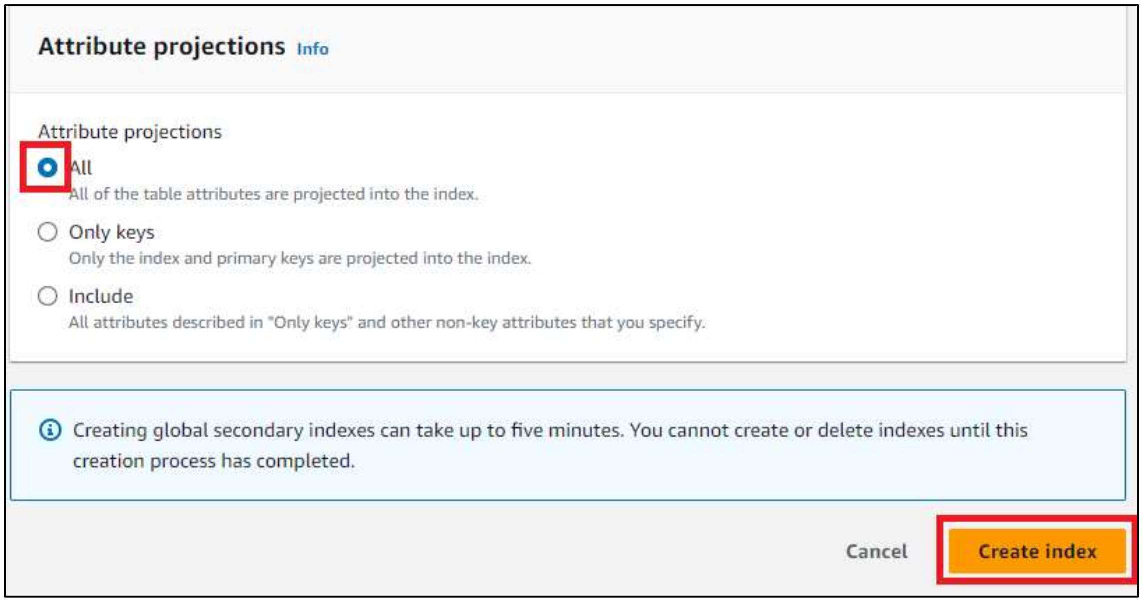
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

○ On

○ Off

Provisioned capacity units

5



### Attribute projections Info

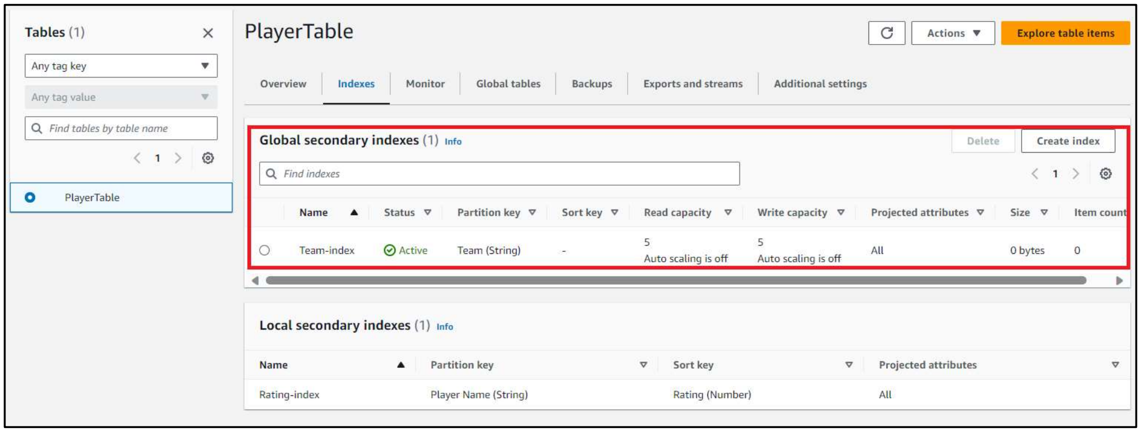
Attribute projections

● All
All of the table attributes are projected into the index.

○ Only keys
Only the index and primary keys are projected into the index.

○ Include
All attributes described in "Only keys" and other non-key attributes that you specify.

ⓘ Creating global secondary indexes can take up to five minutes. You cannot create or delete indexes until this creation process has completed.



Tables (1)

Any tag key

Any tag value

Find tables by table name

○ PlayerTable

PlayerTable

Actions ▼ | Explore table items

Overview | Indexes | Monitor | Global tables | Backups | Exports and streams | Additional settings

Global secondary indexes (1) Info

Delete | Create index

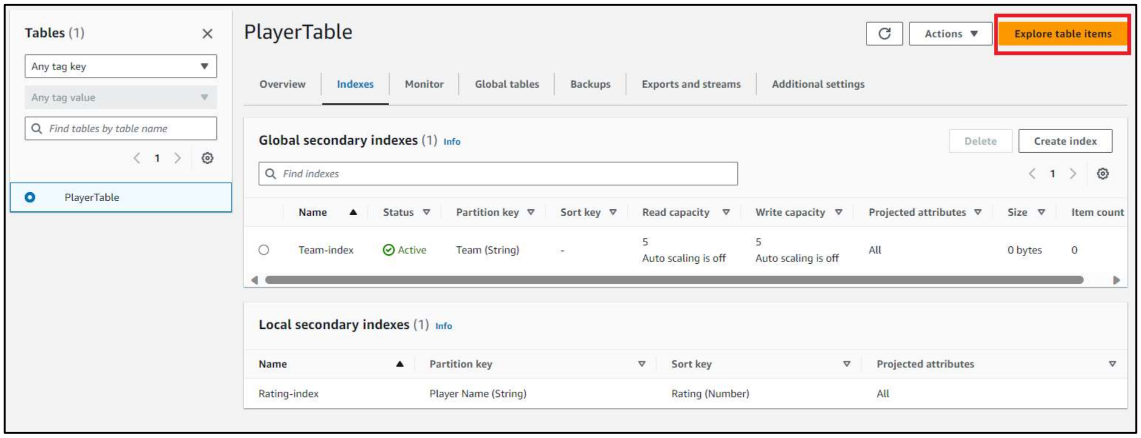
Find indexes

| | Name ▲ | Status | Partition key | Sort key | Read capacity | Write capacity | Projected attributes | Size | Item count |
|---|---|---|---|---|---|---|---|---|---|
| ○ | Team-index | ✓ Active | Team (String) | - | 5<br>Auto scaling is off | 5<br>Auto scaling is off | All | 0 bytes | 0 |

Local secondary indexes (1) Info

| Name ▲ | Partition key | Sort key | Projected attributes |
|---|---|---|---|
| Rating-index | Player Name (String) | Rating (Number) | All |

## Query Global Secondary Index:



Tables (1)

Any tag key

Any tag value

Find tables by table name

○ PlayerTable

PlayerTable

Actions ▼ | Explore table items

Overview | Indexes | Monitor | Global tables | Backups | Exports and streams | Additional settings

Global secondary indexes (1) Info

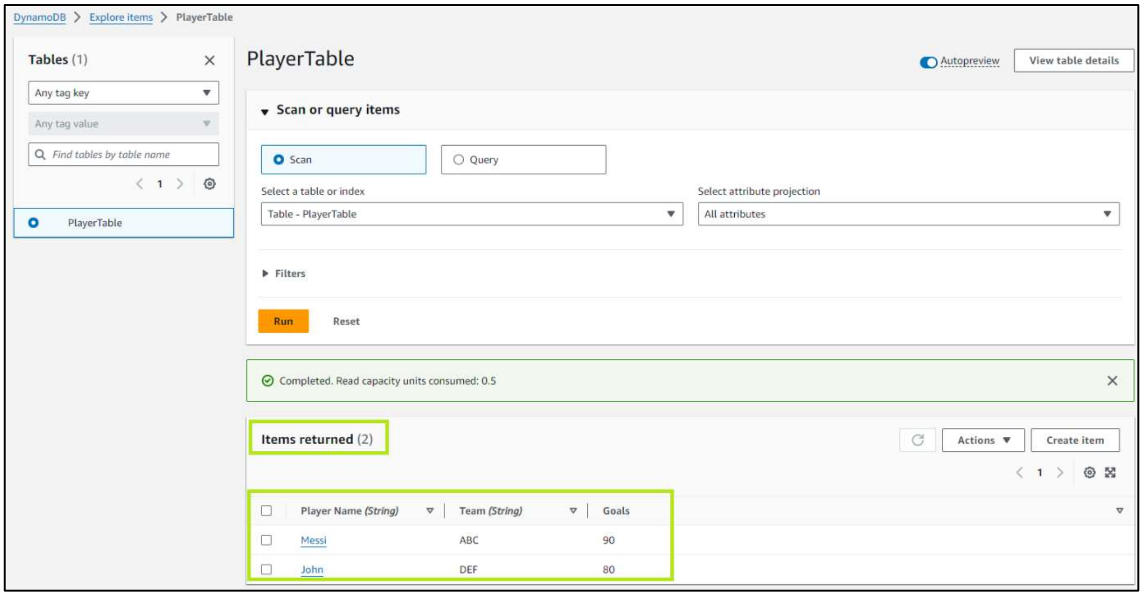
Delete | Create index

Find indexes

| | Name ▲ | Status | Partition key | Sort key | Read capacity | Write capacity | Projected attributes | Size | Item count |
|---|---|---|---|---|---|---|---|---|---|
| ○ | Team-index | ✓ Active | Team (String) | - | 5<br>Auto scaling is off | 5<br>Auto scaling is off | All | 0 bytes | 0 |

Local secondary indexes (1) Info

| Name ▲ | Partition key | Sort key | Projected attributes |
|---|---|---|---|
| Rating-index | Player Name (String) | Rating (Number) | All |



DynamoDB > Explore items > PlayerTable

Tables (1)

Any tag key

Any tag value

Find tables by table name

○ PlayerTable

PlayerTable

Autopreview | View table details

▼ Scan or query items

● Scan ○ Query

Select a table or index: Table - PlayerTable

Select attribute projection: All attributes

▶ Filters

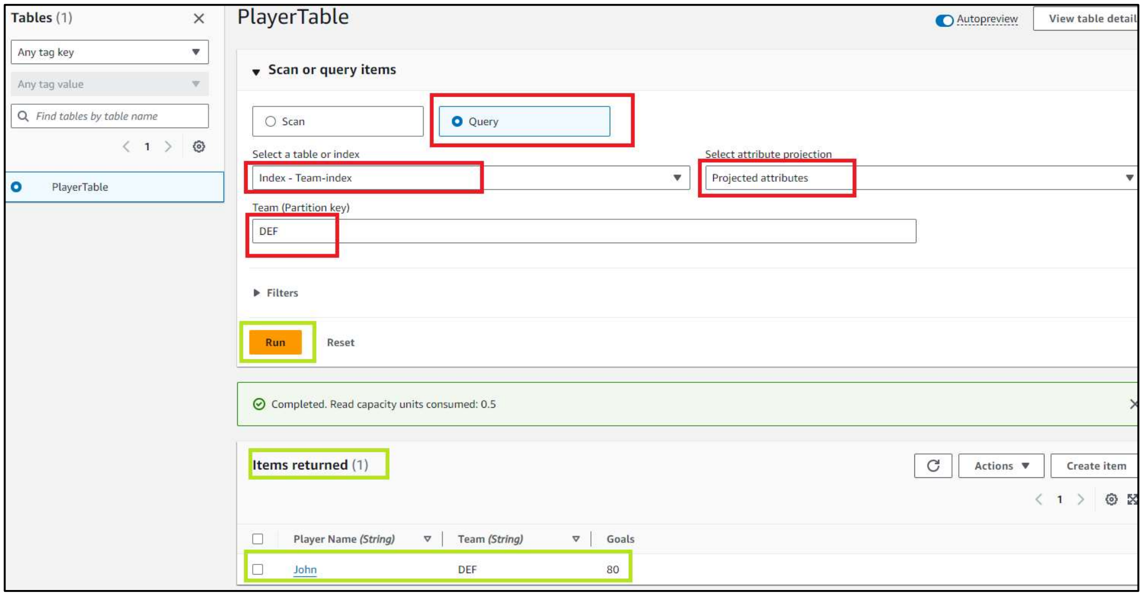
Run | Reset

✓ Completed. Read capacity units consumed: 0.5

Items returned (2)

Actions ▼ | Create item

| □ | Player Name (String) | Team (String) | Goals |
|---|---|---|---|
| □ | Messi | ABC | 90 |
| □ | John | DEF | 80 |



PlayerTable

Scan or query items

○ Scan ● Query

Select a table or index: Index - Team-index

Select attribute projection: Projected attributes

Team (Partition key): DEF

Completed. Read capacity units consumed: 0.5

Items returned (1)

| □ | Player Name (String) | Team (String) | Goals |
|---|---|---|---|
| □ | John | DEF | 80 |