



Query Data with DynamoDB



Saish Nar

```
[cloudshell-user@ip-10-134-42-113 nextworksampleddata]$ aws dynamodb get-item \
>   --table-name ContentCatalog \
>   --key '{"Id":{"N":"101"}}' \
>   --consistent-read \
>   --projection-expression "Title, ContentType, Services" \
>   --return-consumed-capacity TOTAL
{
    "ConsumedCapacity": {
        "TableName": "ContentCatalog",
        "CapacityUnits": 1.0
    }
}
```



Saish Nar
NextWork Student

NextWork.org

Introducing Today's Project!

What is Amazon DynamoDB?

Amazon DynamoDB is a non relational AWS database service. DynamoDB stored data in tables, where data is organised using items and attributes.

How I used Amazon DynamoDB in this project

In today's project I used DynamoDB to create and load tables, ran queries over the AWS CLI and the console. I also processed transactions (i.e. ran multiple operations as one) to manage related data.

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

One thing I didn't expect in this project is that DynamoDB queries have to use the partition key. In scenarios where we want to find data using only other attributes, I run into an error.

This project took me...

This project took me 60 minutes to complete.

Querying DynamoDB Tables

A partition key is the main tag/id that DynamoDB will use to partition its table's items. Every item must have a partition key, and the partition key's value must be unique unless a table also has a sort key.

A sort key is an optional second filter that DynamoDB can use to search for specific items. If a table has a sort key, then each item's primary key is made of its partition key + sort key i.e. partition keys can be the same if sort keys are not.

▼ Scan or query items

Scan Query

Select a table or index Select attribute projection

Id (Partition key)

CommentDateTime (Sort key)
 Sort descending

▶ Filters



Limits of Using DynamoDB

I ran into an error when I queried for all comments made by a specific user. This was because queries MUST use the table's partition key, and the PostedBy attribute (which I used to query my data) was not the partition key.

Insights we could extract from our Comment table includes all comments left on a single post, and even sorted/filtered by comment date/time. Insights we can't easily extract from the Comment table includes all comments made by a specific user.

The screenshot shows the AWS DynamoDB "Scan or query items" interface. The "Query" tab is selected. The "Table - Comment" is chosen for the table. Under "Id (Partition key)", there is a field labeled "Enter partition key value" which is empty, highlighted with a red border, and accompanied by an error message: "The partition key filter cannot be empty." The "CommentDateTime (Sort key)" section is also visible, showing a "Greater than" dropdown and a "Sort descending" checkbox, both of which are currently inactive.



Running Queries with CLI

A query I ran in CloudShell was aws-dynamodb-get-item. This query will grab an item that's from the table and with the partition key value that I described in the command.

Query options I could add to my query are consistent-read (which provides a strongly consistent read), projection-expression (i.e. only return certain attributes) and returned-consumed-capacity (which returns the number of capacity units consumed).

```
[cloudshell-user@ip-10-134-42-113 nextworksampleddata]$ aws dynamodb get-item \
>   --table-name ContentCatalog \
>   --key '{"Id":{"N":"101"}}' \
>   --consistent-read \
>   --projection-expression "Title, ContentType, Services" \
>   --return-consumed-capacity TOTAL
{
    "ConsumedCapacity": {
        "TableName": "ContentCatalog",
        "CapacityUnits": 1.0
    }
}
```



Saish Nar
NextWork Student

NextWork.org

Transactions

A transaction is a group of operations that are bundled together and get executed as one. For the transaction to succeed, all operations within that transaction must process successfully.

I ran a transaction using AWS CLI commands run on AWS CloudShell. This transaction did two things: first, it added a new item to a table (Comment); then, the transaction updated an item in the Forum table (Events' Comment count increased by 1).

```
[cloudshell-user@ip-10-134-42-113 nextworksampleddata]$ aws dynamodb transact-write-items --client-request-token TRANSACTION1 --transact-items '[  
>     {  
>         "Put": {  
>             "TableName" : "Comment",  
>             "Item" : {  
>                 "Id" : {"S": "Events/Do a Project Together - NextWork Study Session"},  
>                 "CommentDateTime" : {"S": "2024-9-27T17:47:30Z"},  
>                 "Comment" : {"S": "Excited to attend!"},  
>                 "PostedBy" : {"S": "User Connor"}  
>             }  
>         },  
>         {  
>             "Update": {  
>                 "TableName" : "Forum",  
>                 "Key" : {"Name" : {"S": "Events"}},  
>                 "UpdateExpression": "ADD Comments :inc",  
>                 "ExpressionAttributeValues" : { ":inc" : {"N" : "1"} }  
>             }  
>         }  
>     }'  
> ]'
```



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

