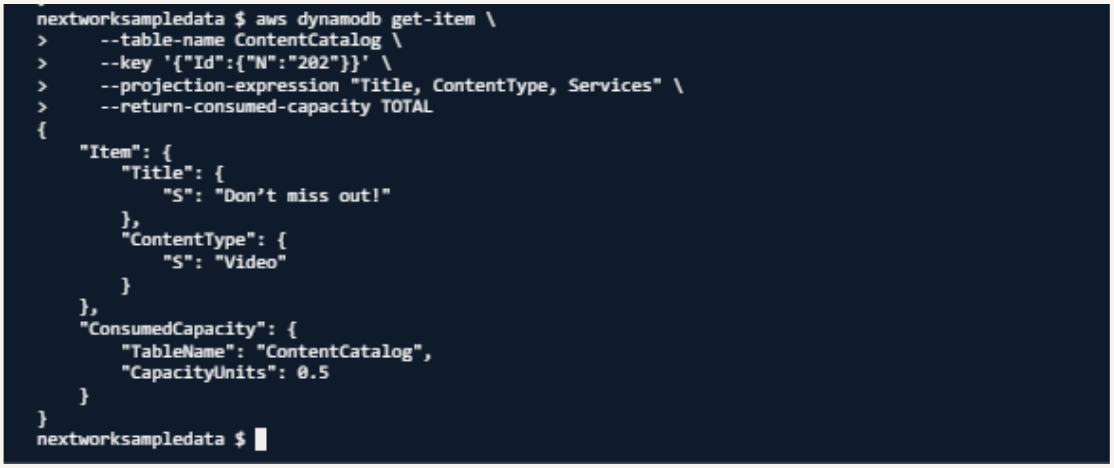
nextwork.org

Query Data with DynamoDB

DI Dineshraj Dhanapathy



DI
Dineshraj Dhanapathy
NextWork Student

Introducing Today's Project!

What is Amazon DynamoDB?

Amazon DynamoDB is a fully managed NoSQL database that offers fast performance, scalability, and flexibility. It is useful for handling high-traffic applications, ensuring low-latency access, and automating scaling.

How I used Amazon DynamoDB in this project

I used Amazon DynamoDB to create tables, load data, run queries, and perform transactions. This helped in managing and retrieving data efficiently while ensuring consistency across multiple tables.

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

One thing I didn't expect in this project was how important data modeling is in DynamoDB. Choosing the right partition and sort keys made a big difference in query performance and data retrieval efficiency.

DI

Dineshraj Dhanapathy
NextWork Student

This project took me...

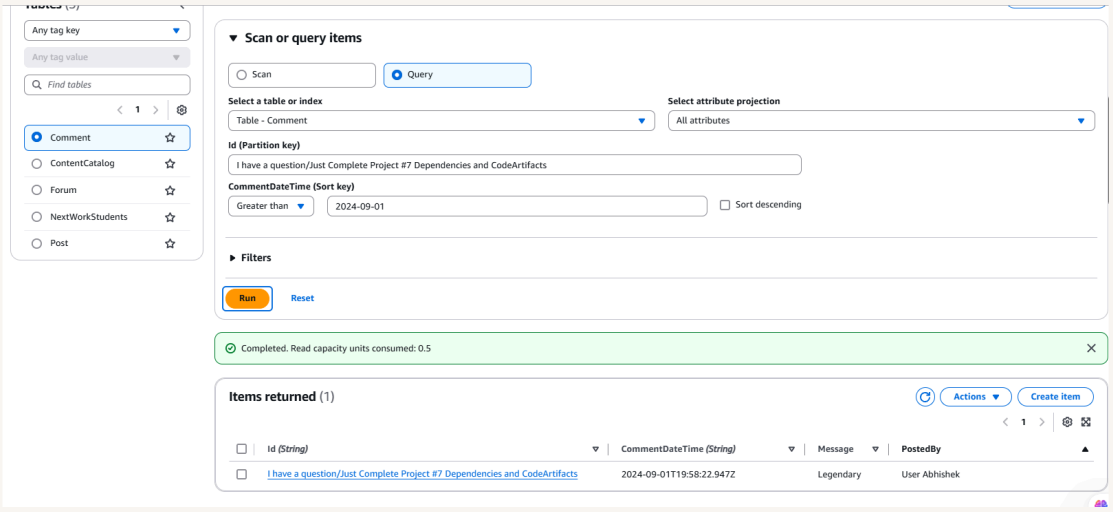
This project took me a few hours to complete, mainly due to setting up DynamoDB tables, running queries, and troubleshooting errors. Understanding partition keys and transactions also took some extra time.

DI
Dineshraj Dhanapathy
NextWork Student
NextWork.org

Querying DynamoDB Tables

A partition key is a primary key in DynamoDB that helps distribute and locate data efficiently. It can have duplicate values, grouping related items together for faster queries and optimized performance.

A sort key is an optional secondary key in DynamoDB that further filters query results after using the partition key. It helps organize and retrieve data efficiently within each partition.

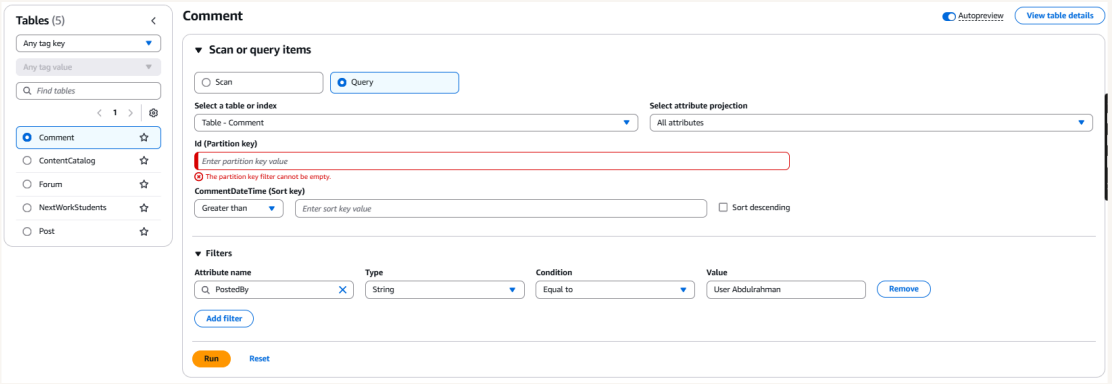

Any tag key
Any tag value
Find tables
Comment
ContentCatalog
Forum
NextWorkStudents
Post
Scan or query items
Scan
Query
Select a table or index
Table - Comment
Select attribute projection
All attributes
Id (Partition key)
I have a question/Just Complete Project #7 Dependencies and CodeArtifacts
CommentDateTime (Sort key)
Greater than
2024-09-01
Sort descending
Filters
Run
Reset
Completed. Read capacity units consumed: 0.5
Items returned (1)
Actions
Create item
Id (String)
CommentDateTime (String)
Message
PostedBy
I have a question/Just Complete Project #7 Dependencies and CodeArtifacts
2024-09-01T19:58:22.947Z
Legendary
User Abhishek

DI
Dineshraj Dhanapathy
NextWork Student

Limits of Using DynamoDB

I ran into an error when I queried for an item without using the required Partition Key (Id). This was because DynamoDB needs the partition key to locate data efficiently, highlighting the importance of data modeling.

Insights we could extract from our Comment table include user interactions, comment frequency, and engagement trends. Insights we can't easily extract include sentiment analysis and contextual relevance without extra processing

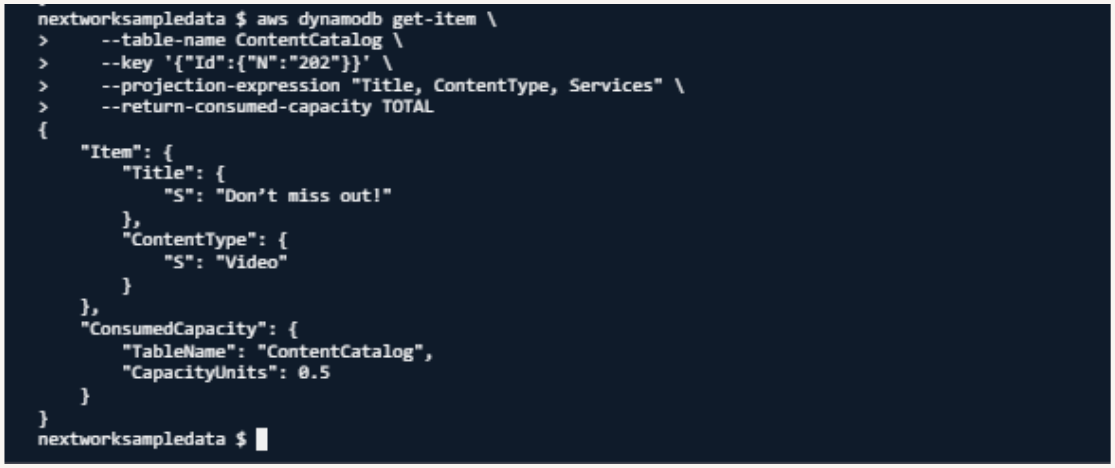


DI
Dineshraj Dhanapathy
NextWork Student

Running Queries with CLI

A query I ran in CloudShell was retrieving an item from the ContentCatalog table using its Id (202) while selecting only Title, ContentType, and Services. This query helps fetch specific attributes efficiently.

Query options I could add to my query are projection-expression, which selects specific attributes, consistent-read, which ensures up-to-date data, and return-consumed-capacity, which tracks query costs.


nextworksampledata $ aws dynamodb get-item \
> --table-name ContentCatalog \
> --key '{"Id":{"N":"202"}}' \
> --projection-expression "Title, ContentType, Services" \
> --return-consumed-capacity TOTAL
{
"Item": {
"Title": {
"S": "Don't miss out!"
},
"ContentType": {
"S": "Video"
}
},
"ConsumedCapacity": {
"TableName": "ContentCatalog",
"CapacityUnits": 0.5
}
}
nextworksampledata $

DI
Dineshraj Dhanapathy
NextWork Student

Transactions

A transaction is a group of operations in DynamoDB that must all succeed, ensuring consistency. If one fails, none apply. This helps update multiple tables reliably, like adding a comment and updating a count.

I ran a transaction using AWS CLI, ensuring multiple operations succeeded together. This transaction did two things: it added a new comment to the Comment table and updated the comment count in the Forum table.

```
nextworksampledata $ 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"}
>             }
>         }
>     },
```

NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects