nextwork.org

Query Data with DynamoDB

Sanjana Tripathy

Sanjana Tripathy
NextWork Student

Introducing Today's Project!

What is Amazon DynamoDB?

Amazon DynamoDB is a fully managed NoSQL database service that offers fast, predictable performance and seamless scalability. It is ideal for applications requiring low-latency data access, high availability, and automatic scaling.

How I used Amazon DynamoDB in this project

I created a DynamoDB table and inserted data from scratch using AWS CloudShell. I queried the table via the console and CloudShell, then executed a transaction to update two tables atomically for consistent data handling.

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

I didn’t expect how seamless and powerful CloudShell would be for managing DynamoDB. Running transactions and interacting with the service entirely from the terminal gave me deeper control and understanding than using the console alone.

Sanjana Tripathy
NextWork Student

This project took me...

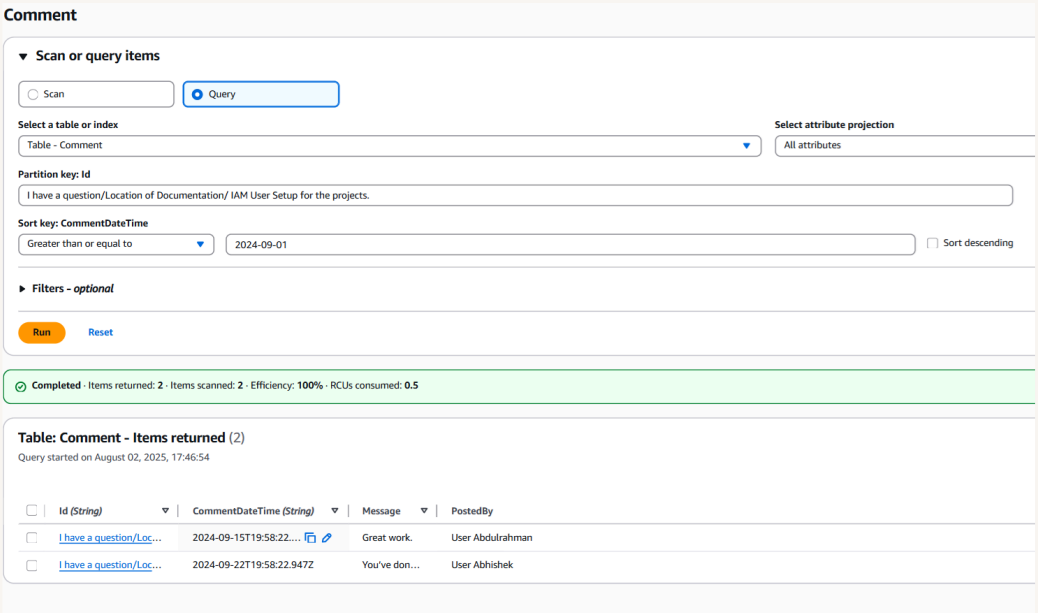
This project took approximately 2 hours to complete.

Sanjana Tripathy
NextWork Student

Querying DynamoDB Tables

A partition key is a required attribute in DynamoDB used to uniquely identify and distribute items across storage partitions. It determines where data is stored and plays a key role in ensuring fast and efficient data access.

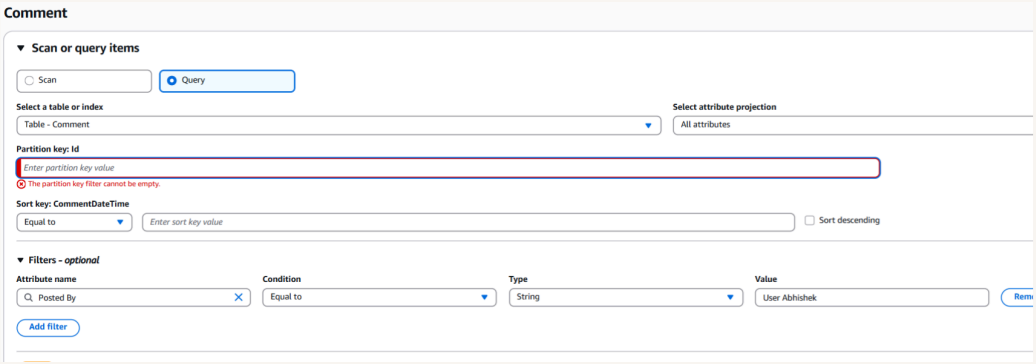
A sort key is an optional secondary attribute used with the partition key to form a composite primary key in DynamoDB. It allows multiple items with the same partition key to be sorted and queried efficiently based on the sort key value.



Limits of Using DynamoDB

I ran into an error when I queried for all comments posted by a single user. This was because when I added the user as the value in the filter, patition key was not used.But partition key is essential to query data in dynamoDB table

I ran into an error when querying all comments by a user because the partition key was set to the comment ID. Since DynamoDB requires the partition key in queries, I couldn't fetch all user comments without redesigning the table structure.



Running Queries with CLI

A query I ran in CloudShell was:aws dynamodb get-item --table-name ContentCatalog --key '{"Id":{"N":"202"}}' --projection-expression "Title, ContentType, Services" --return-consumed-capacity TOTAL.This query retrieves selected attributes.

Query options I could add to my query are --consistent-read for the most up-to-date data, --projection-expression to return only specific attributes, and --return-consumed-capacity to see how much read capacity the request used.

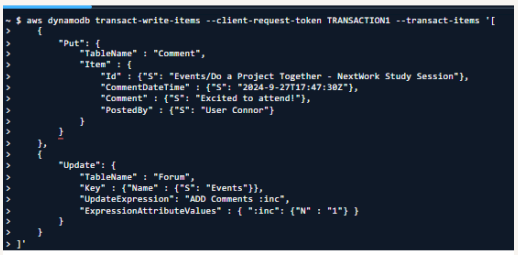
```
~ $ aws dynamodb get-item \
>     --table-name ContentCatalog \
>     --key '{"Id":{"N":"202"}}' \
>     --projection-expression "Title, ContentType, Services" \
>     --return-consumed-capacity TOTAL
{
    "Item": {
        "Title": {
            "S": "Don’t miss out!"
        },
        "ContentType":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

Sanjana Tripathy
NextWork Student

Transactions

"A transaction is a set of read or write operations in DynamoDB that are executed together atomically. If any operation fails, all changes are rolled back, ensuring consistency and integrity across items and tables."

"I ran a transaction using DynamoDB TransactWriteItems. This transaction added Connor's comment to the Comment table and incremented the comment count in the Forum table, ensuring both updates succeed together."



The place to learn & showcase your skills

Check out nextwork.org for more projects