nextwork.org

Query Data with DynamoDB

Louis Moyo

Louis Moyo
NextWork Student

Introducing Today's Project!

What is Amazon DynamoDB?

Amazon DynamoDB is a fully managed NoSQL (key-value) database from AWS. It's useful because it's serverless, auto-scales, gives millisecond reads/writes, has flexible schemas, and you only pay for what you use.

How I used Amazon DynamoDB in this project

In today's project, I used Amazon DynamoDB to store and query app data. I created four tables (ContentCatalog, Forum, Post, Comment) with CloudShell/CLI, loaded sample JSON, and explored items in the console. I ran queries with partition/sort keys and get-item, saw why queries need the partition key, and finished with a transaction that added a new Comment and atomically bumped the Comments counter on the related Forum.

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

On thing that I didn't expect in this project is queries must use the partition key - filters alone won't work. Also, "strongly consistent" reads cost more than eventual reads.

Louis Moyo
NextWork Student

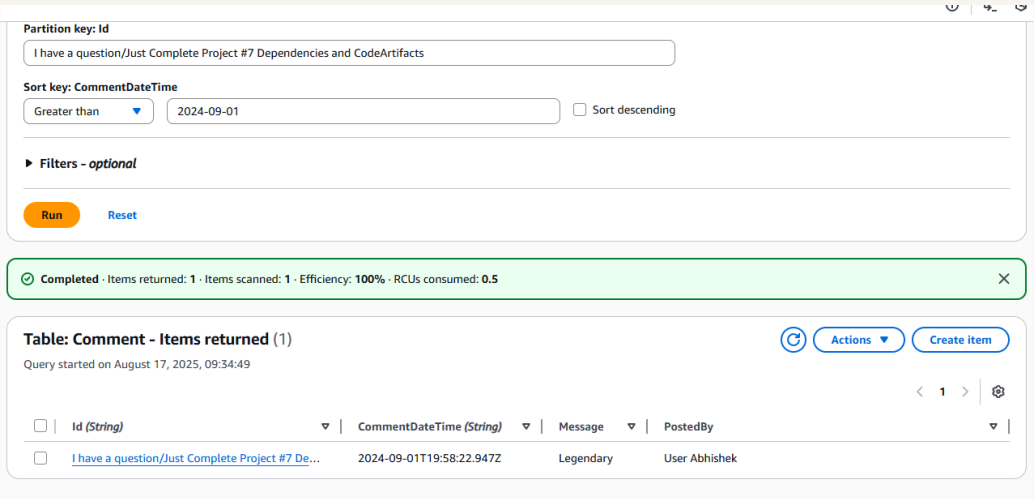
This project took me...

This project took me 2 hours.

Querying DynamoDB Tables

A partition key is the main ID DynamoDB uses to find and group items. If there's no sort key, it must be unique per item.

A sort key is an optional second ID that orders items with the same partition key (e.g., by date), letting you store many items under one key and query by ranges.

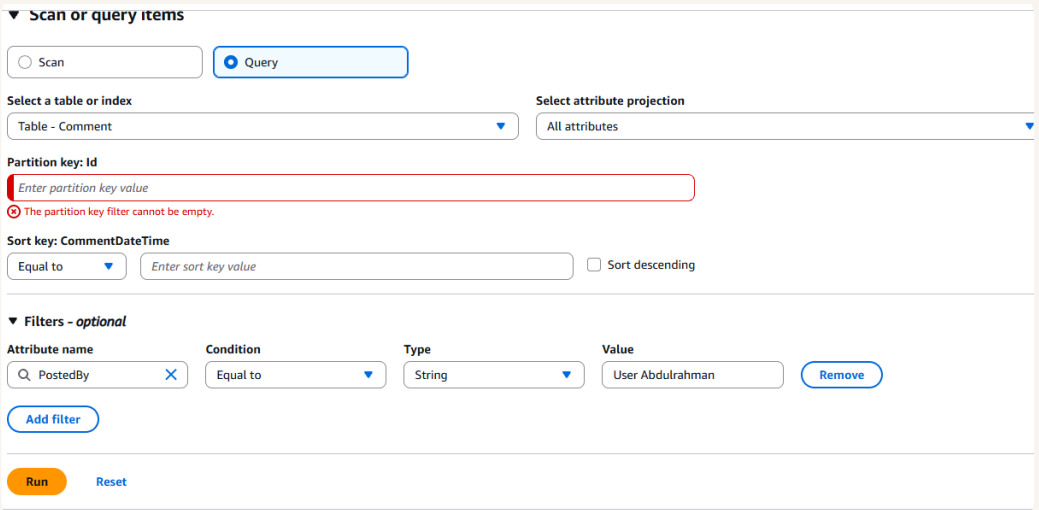


Louis Moyo
NextWork Student

Limits of Using DynamoDB

I ran into an error when I queried for all comments by “User Abdulrahman”. This was because I cleared the partition key and tried to filter only on PostedBy.

Insights we could extract from our Comment table include: comments within each thread (by Id), newest/oldest comments using the date, and when threads are most active. Insights we can't easily extract from the Comment table include: all comments by a specific user, top commenters across threads, or keyword searches - because the keys are set by thread, not by user/content.



L
Louis Moyo
NextWork Student

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 will fetch the item with Id 202, show only Title, ContentType, Services, and include a note about how much capacity the read used.

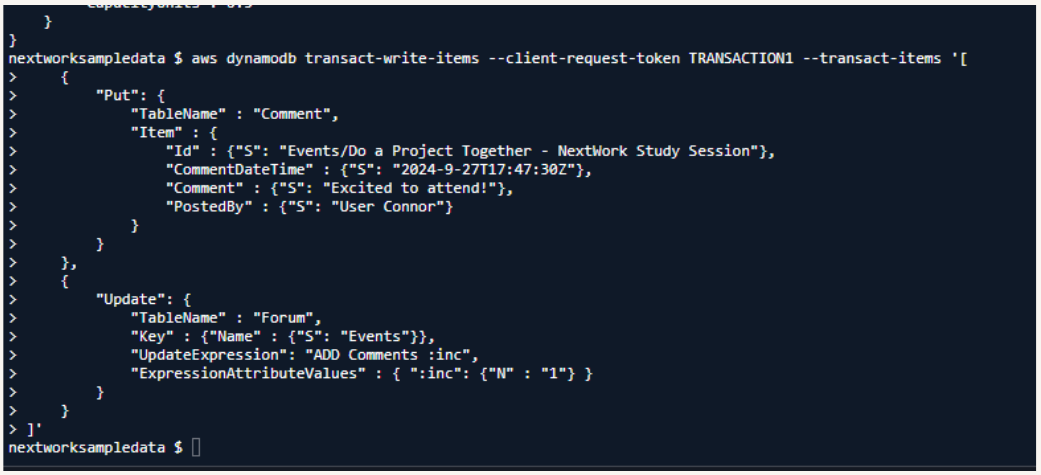
Query options I could add to my query are: --consistent-read → get the latest version (strong consistency, costs more). --projection-expression → return only the fields I ask for. --return-consumed-capacity TOTAL → show the capacity used by the request.

Louis Moyo
NextWork Student

Transactions

A transaction is a bundled set of database actions that succeed or fail together. In DynamoDB, a transaction guarantees all-or-nothing updates - so related writes stay consistent even if something goes wrong.

I ran a transaction using aws dynamodb transact-write-items. This transaction did two things in one atomic step: Put a new item in the Comment table (the new comment). Update the Forum item for Name = "Events" to increment its Comments count by 1. If either step failed, neither change was saved.



The place to learn & showcase your skills

Check out nextwork.org for more projects