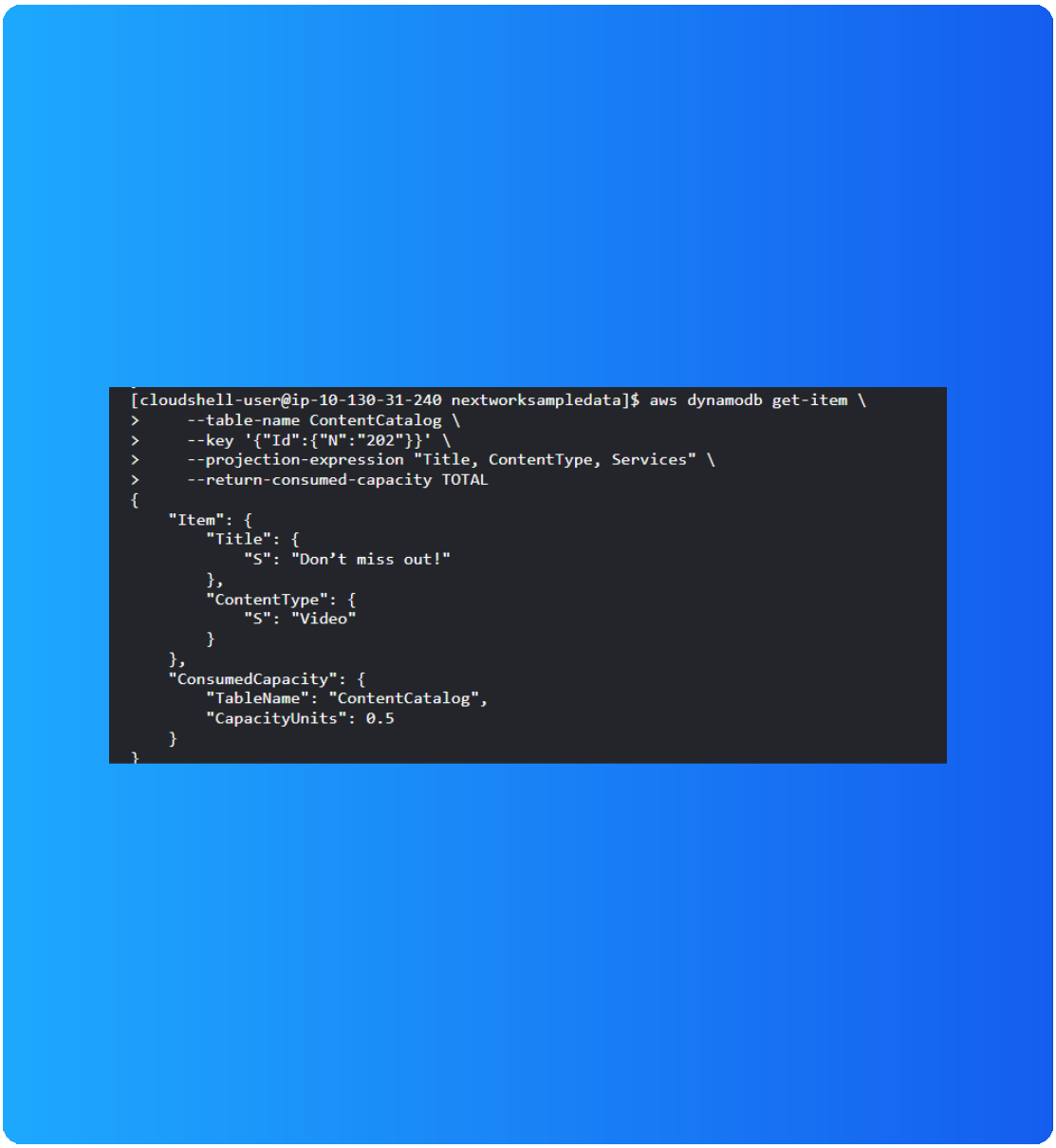
NextWork.org

Query Data With DynamoDB

Praneeth Bhandwalkar



Praneeth Bhandwalkar
NextWork Student

Introducing Today's Project!

What is Amazon DynamoDB?

Amazon DynamoDB is a fully managed NoSQL database service designed for fast, predictable performance with single-digit millisecond response times. It supports both key-value and document data structures, making it versatile for various applications.

How I used Amazon DynamoDB in this project

In this project, I used DynamoDB to create a table and load data into the tables, run "Queries" using the AWS CLI, and also used transactions for updating the tables.

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

The only thing I didn't expect in this project when querying items we need to use an ID (Partition key) without that error occurred. another thing, I didn't expect was the AWS CLI used for updating the table using “transactions” which is exclusive to AWS CLI because transactions don't exist in the AWS Management Console.

This project took me...

It took me almost 65 minutes to complete.

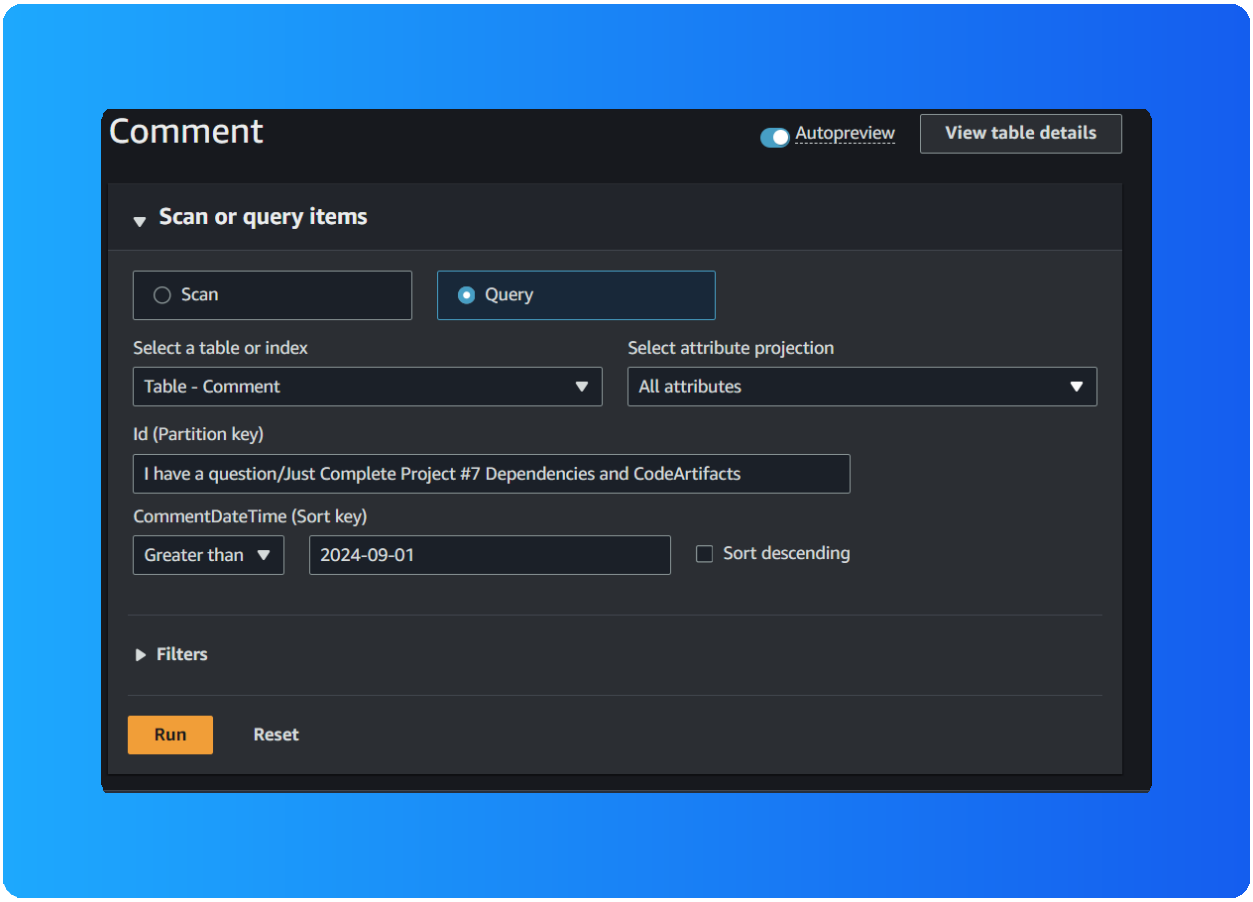
Praneeth Bhandwalkar
NextWork Student

NextWork.org

Querying DynamoDB Tables

A partition key is used to filter that DynamoDB will use to split up and find data.

A sort key is a secondary key used to filter your query results again! Sort keys work after the partition key.

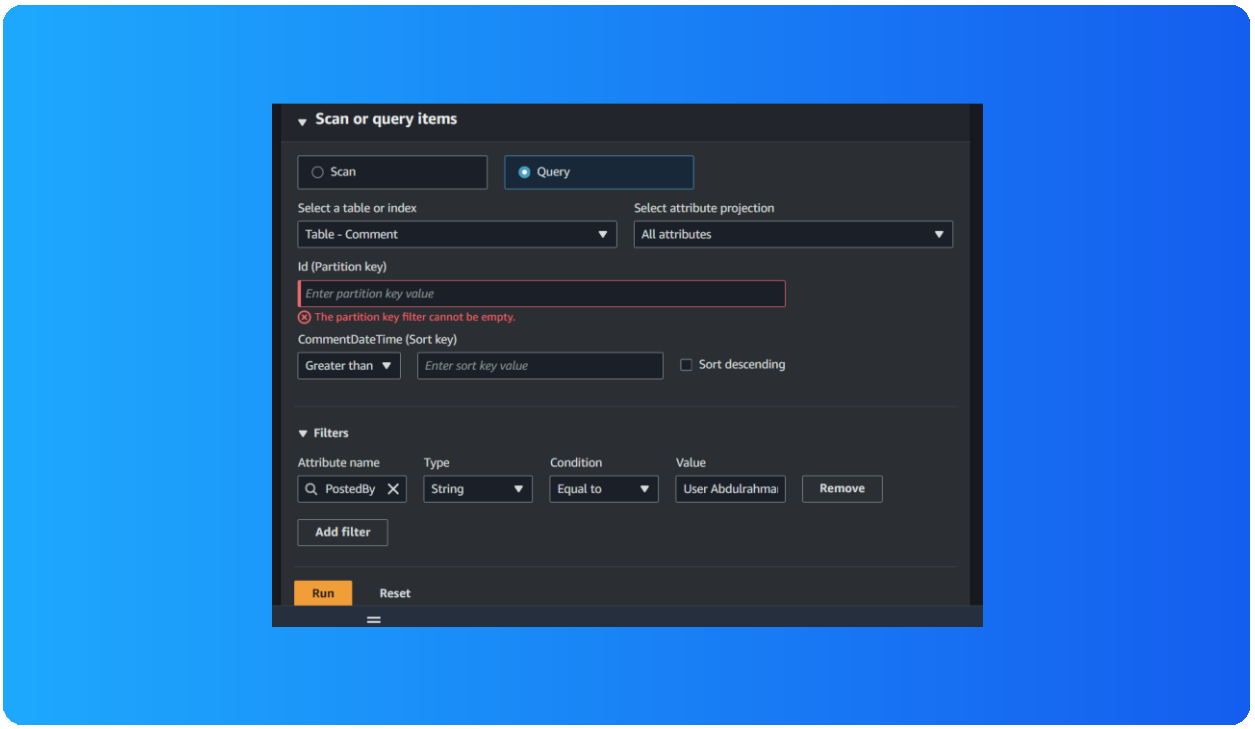


Praneeth Bhandwalkar
NextWork Student

Limits of Using DynamoDB

I ran into an error when I queried for all comments made by a user. This was because tables should use the partition keys when querying the items.

Insights we could extract from our Comment table include all the comments on the post. Insights we can't easily extract from the Comment table include all the comments made by a user.



Praneeth Bhandwalkar
NextWork Student

Running Queries with CLI

A query I ran in CloudShell was aws dynamodb get-item. This query will make a single item from a DynamoDB table and how the items belong to the table with the key which tells its ID and a Number.

Query options I could add to my query are consistent-read, projection-expression, and return-consumed-capacity query options that tell DynamoDB exactly how we want it to get the item.

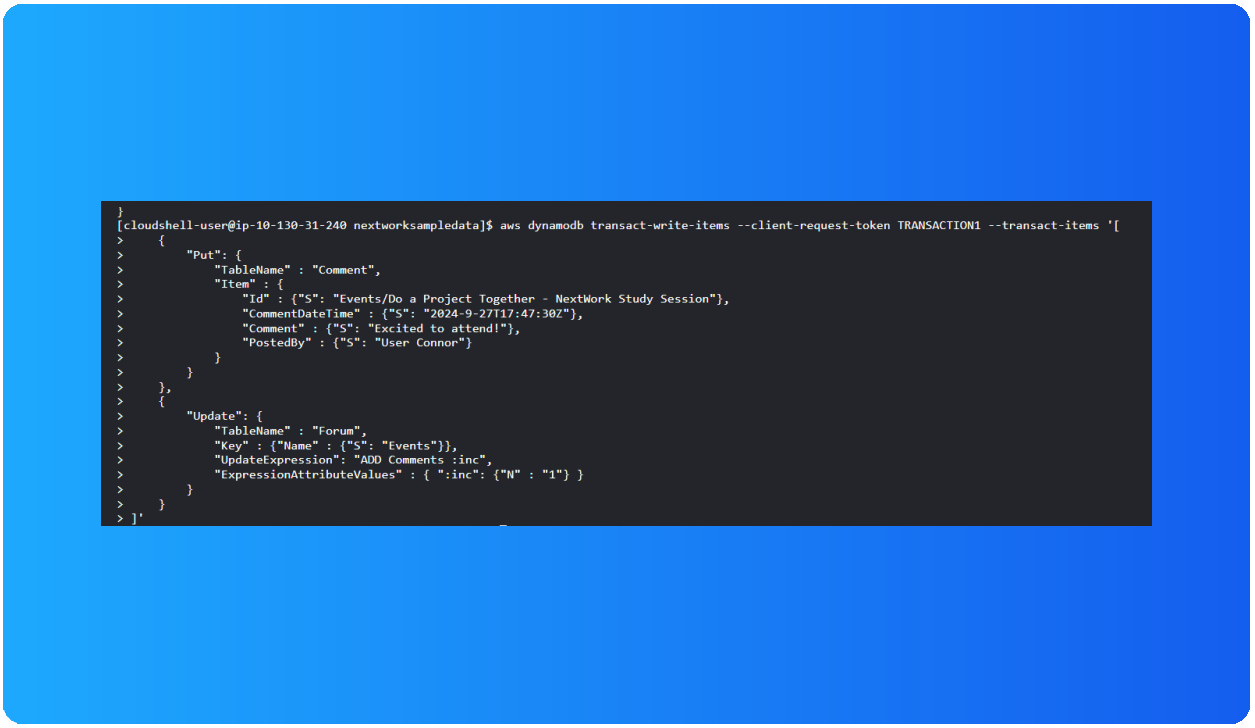
```
[cloudshell-user@ip-10-130-31-240 nextworksampledata]$ aws dynamodb get-item \
>     --table-name ContentCatalog \
>     --key '{"Id":{"N":"202"}}' \
>     --projection-expression "Title, ContentType, Services" \
>     --return-consumed-capacity TOTAL
{
    "Item": {
        "Title": {
            "S": "Don't miss out!"
        },
        "ContentType": {
            "S": "Video"
        }
    },
    "ConsumedCapacity": {
        "TableName": "ContentCatalog",
        "CapacityUnits": 0.5
    }
}
```

Praneeth Bhandwalkar
NextWork Student

Transactions

A transaction is a command that is a group of operations that all have to succeed if any of the operations in the group fails, none of the changes get applied.

I ran a transaction using aws dynamodb transact-write-items. This transaction did two things adding a new comment to the event and updating both tables which are the Comment table and the Forum table.



```
}
[cloudshell-user@ip-10-130-31-240 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"}
>             }
>         }
>     },
>     {
>         "Update": {
>             "TableName" : "Forum",
>             "Key" : {"Name" : {"S": "Events"}},
>             "UpdateExpression": "ADD Comments :inc",
>             "ExpressionAttributeValues" : { ":inc": {"N" : "1"} }
>         }
>     }
> ]'
```

Everyone should be in a job they love.

Check out nextwork.org for more projects