



NextWork.org

Load and Query DynamoDB Tables



ajv.singh3107@gmail.com

```
aws dynamodb get-item \
>   --key '{"Id":{"N":"20211"}' \
>   --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
  }
}
[cloudshell]:user@ip-10-130-2-211 ~]$ aws dynamodb get-item \
[cloudshell]:user@ip-10-130-2-211 ~$[cloudshell]:user@ip-10-130-2-211 ~$[cloudshell]:user@ip-10-130-2-211 ~$[cloudshell]:user@ip-10-130-2-211 ~$
```

Introducing Today's Project!

What is Amazon DynamoDB?

Amazon DynamoDB is a fully managed NoSQL database service that offers fast performance and scalability. It's ideal for apps needing low-latency data access, like mobile and web apps, without the overhead of managing database infrastructure.

How I used Amazon DynamoDB in this project

In today's project, I used Amazon DynamoDB to create tables, load data, and run queries. I also utilized transactions to update multiple tables at once, enabling efficient data management and ensuring consistency across related items.

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

One thing I didn't expect in this project was how easily I could manage and query related data across multiple tables using transactions. It simplified the process of updating and ensuring consistency between different data points in DynamoDB.

This project took me...

This project took me about 2 hours. Setting up the DynamoDB tables, running queries, and experimenting with transactions in AWS CloudShell were the main tasks.

Querying DynamoDB Tables

A partition key is the primary key attribute in DynamoDB that uniquely identifies each item in a table. It determines how the data is distributed across partitions for efficient access and scaling, ensuring fast data retrieval based on the key's value.

A sort key is an optional attribute in DynamoDB that allows you to sort items within the same partition key. It enables more precise querying, helping to retrieve items in a specific order or filter based on the sort key's value.

The screenshot shows the AWS DynamoDB console's "Scan or query items" interface. The "Query" tab is selected. A table named "Table - Comment" is chosen. The "Id (Partition key)" field contains the value "I have a question/Just Complete Project #7 Dependencies and CodeArtifacts". The "CommentDateTime (Sort key)" field has an "Equal to" dropdown set to "2024-09-01" and a "Sort descending" checkbox checked. The "Select attribute projection" dropdown is set to "All attributes". At the bottom, there are "Run" and "Reset" buttons.

Limits of Using DynamoDB

I ran into an error when I queried for... This was because the partition key filter cannot be empty. DynamoDB requires a partition key value to specify which partition to query. Without a partition key, the query cannot be executed, resulting in error

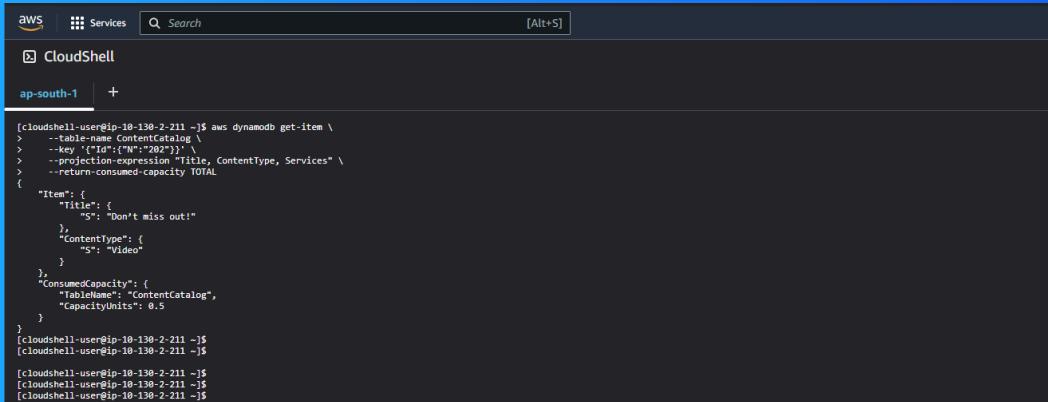
Insights we could extract from our Comment table includes comment counts, user engagement, and comment trends over time. Insights we can't easily extract include sentiment analysis or post-interaction patterns without additional processing.

The screenshot shows the AWS DynamoDB Query interface. The top navigation bar has 'Scan or query items' and 'Query' selected. Under 'Select a table or index', 'Table - Comment' is chosen. Under 'Select attribute projection', 'All attributes' is selected. In the 'Id (Partition key)' section, there is a text input field with 'Enter partition key value' placeholder text. A red error message 'The partition key filter cannot be empty.' is displayed below the input field. Below this, under 'CommentDateTime (Sort key)', there is a dropdown set to 'Greater than' and a text input field for 'Enter sort key value'. A checkbox for 'Sort descending' is present but unchecked. At the bottom, there is a 'Filters' section with a single filter for 'PostedBy': Attribute name 'PostedBy', Type 'String', Condition 'Equal to', Value 'User Abdulrahman', and a 'Remove' button. Below the filters is a 'Run' button and a 'Reset' button.

Running Queries with CLI

This query will return the "Title," "ContentType," and "Services" attributes for the item with "Id" 202 and show the consumed capacity.

Query options I could add to my query are: 1. `--projection-expression`: Limits returned attributes. 2. `--consistent-read`: Ensures fresh data. 3. `--return-consumed-capacity`: Shows consumed read capacity.



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal prompt is [cloudshell-user@ip-10-130-2-211 ~]\$ and the user is executing an AWS CLI command to query a DynamoDB table named 'ContentCatalog'. The command includes projection expressions for 'Title', 'ContentType', and 'Services', and specifies the consumption of read capacity. The output shows the retrieved item with its attributes and the consumed capacity information.

```
[cloudshell-user@ip-10-130-2-211 ~]$ 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
  }
}
[cloudshell-user@ip-10-130-2-211 ~]$ [cloudshell-user@ip-10-130-2-211 ~]$ [cloudshell-user@ip-10-130-2-211 ~]$ [cloudshell-user@ip-10-130-2-211 ~$]
```



ajv.singh3107@gmail.com
NextWork Student

NextWork.org

Transactions

A transaction is a sequence of database operations that are treated as a single unit. This means that either all operations within the transaction are completed successfully, or they are rolled back to their original state, ensuring data consistency.

I ran a transaction using AWS CLI's transact-write-items command. This transaction did two things: it updated two DynamoDB tables simultaneously, ensuring that both operations either succeed or fail together, maintaining data integrity across the tables.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

