Cloud Computing Exercise #13

Creating a DynamoDB Table with a Composite Key and a Secondary Index

A. Preparation

1. Sign in to your AWS account as the non-root admin user.

B. Create a DynamoDB table

1. Go to the DynamoDB dashboard and create a table called “Songs” that will hold information about the performing artist, the song’s title, the song’s length and other meta-information. The primary key of the table should consist of a partition key and a sort key. The partition key’s attribute name should be “artist” (the performing artist’s name) and have string type. The sort key’s attribute name should be “title” (the title of the song) and have string type. Click on the “Customize settings” and scroll down to the “Read/write capacity settings**”** Turn off the read/write capacity auto-scaling functionality.
2. Add the “length” attribute (the songs’ durations in minutes) as a secondary index to the table to allow for range queries based on the lengths of the songs. Scroll down to the “Secondary indices” section and add a local secondary index (NOT a global secondary index) to the table with sort key name “length” and type number (you can use the default index name). In the “Attribute projections” sub-section, select the “Only keys” option. This will copy only the primary key (partition key and sort key) into the index data structure. Then, create the table.

C. Add items to the table

1. Use the Management Console to add the following set of items to the Songs table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| artist  [string]  (part. key) | title  [string]  (sort key) | length  [number]  (secondary sort key) | isFavorite  [boolean] | metaDate  [map] |
| The Beatles | Come together | 4.19 | false | N/A |
| The Beatles | Here comes the sun | 3.06 | false | N/A |
| The Beatles | Hey Jude | 7.11 | true | Please see the text below |
| Queen | Bohemian Rhapsody | 6.00 | true | N/A |
| Queen | We will rock you | 2.02 | true | N/A |
| Queen | Radio ga ga | 5.48 | false | N/A |

When adding attributes for the song “Hey Jude” in the item editor, click on the “JSON” button in the upper right corner to see how the data is actually formatted in JSON (this is the real internal representation). Use the JSON format to enter the attributes for this song, including the metaData attribute, which is defined as:

{

"artist": {

"S": "The Beatles"

},

"title": {

"S": "Hey Jude"

},

"isFavorite": {

"BOOL": false

},

"length": {

"N": "7.11"

},

"metaData": {

"M": {

"genre": {

"S": "Rock"

},

"location": {

"S": "c:\\Music\\Beatles\\Hey\_Jude.mp3"

},

"writer": {

"S": "John Lennon"

}

}

}

}

You can toggle between the web form view and the JSON format view to see how they correspond to each other. The DynamoDB’s type notations are as follows: “S” – string, “N” – number, “BOOL” – boolean, “M” – map (JSON object). By toggling the “View DynamoDB JSON” switch, you can see and edit a simpler JSON version that does not have the attribute type information, only the attribute names and values.

It should look something like this: A white rectangular object with a black line

Description automatically generated

D. Query the table

1. After adding all items, expand the “Songs” section on the right hand side and run the following queries:
   * Return all Queen songs whose titles begin with the letter “B”. The ”Table or index” parameter should be set to the table’s name (Songs), the partition key should be set to “Queen”, and the sort key condition should be set to “begins with B”. There should be one such result: “Bohemian Rhapsody”. This is the case when we have used the sort key part of the primary key (the title attribute) to run a range query on the song titles.

A screenshot of a computer

Description automatically generated

* + Return all Beatles songs that are longer than 4.00 (minutes). The “Table or index” parameter should set to the name of the local secondary index (index-length), the partition key should be set to “The Beatles”, and the secondary sort key condition should be set to “greater than 4”. There should be two such results: “Come together” and “Hey Jude”. This is the case when we have used the sort key of the local secondary index (the length attribute) to perform a range query on the length of the songs.

The output should look something like this:

A screenshot of a computer

Description automatically generated

E. Clean up after yourself

1. Delete the Songs table and log out of AWS.