

DBT Assignment-4

ASHWIN KRISHNA P

PES1201801465

SEM: 5 , SECTION: F

DynamoDB (Key Value Database)

Creating a table called Music, and specifying attributes:-

```
Command Prompt
C:\Users\ashwi>aws dynamodb create-table --table-name Music --attribute-definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S --key-schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5
{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "Music",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "CREATING",
    "CreationDateTime": "2020-11-12T12:25:10.497000+05:30",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:ap-south-1:207477500461:table/Music",
    "TableId": "e32e141f-69ea-44b4-a870-44daf4ecb056"
  }
}
```

Now the Music table has been created successfully.

The screenshot shows the AWS Management Console interface for DynamoDB. The left sidebar contains navigation links for Services, Tables, Backups, Reserved capacity, Preferences, DAX, and Events. The main content area displays the 'Music' table, which is in an 'Active' state. The table's configuration is shown in a table with columns: Name, Status, Partition key, Sort key, Indexes, Total read capacity, and Total write capacity. The 'Music' table has a Partition key of 'Artist (String)' and a Sort key of 'SongTitle (String)'. The 'Total read capacity' is 10 and the 'Total write capacity' is 5.

| Name | Status | Partition key | Sort key | Indexes | Total read capacity | Total write capacity |
|-------|--------|-----------------|--------------------|---------|---------------------|----------------------|
| Music | Active | Artist (String) | SongTitle (String) | 0 | 10 | 5 |

Command to Describe the table:-

```
Command Prompt
C:\Users\ashwi>aws dynamodb describe-table --table-name Music
{
  "Table": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "Music",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "ACTIVE",
    "CreationDateTime": "2020-11-12T12:25:10.497000+05:30",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:ap-south-1:207477500461:table/Music",
    "TableId": "e32e141f-69ea-44b4-a870-44daf4ecb056"
  }
}
```

Insert1.json and insert2.json files

```
item_insert1 - Notepad
File Edit Format View Help
{
  "Artist": {"S": "Ace of Base"},
  "SongTitle": {"S": "Happy Nations"},
  "AlbumName": {"S": "Greatest Hits"}
}
```

```
item_insert2 - Notepad
File Edit Format View Help
{
  "Artist": {"S": "The Chainsmokers"},
  "SongTitle": {"S": "Something Just Like This"},
  "AlbumName": {"S": "Memories"}
}
```

Inserting the above 2 items:-

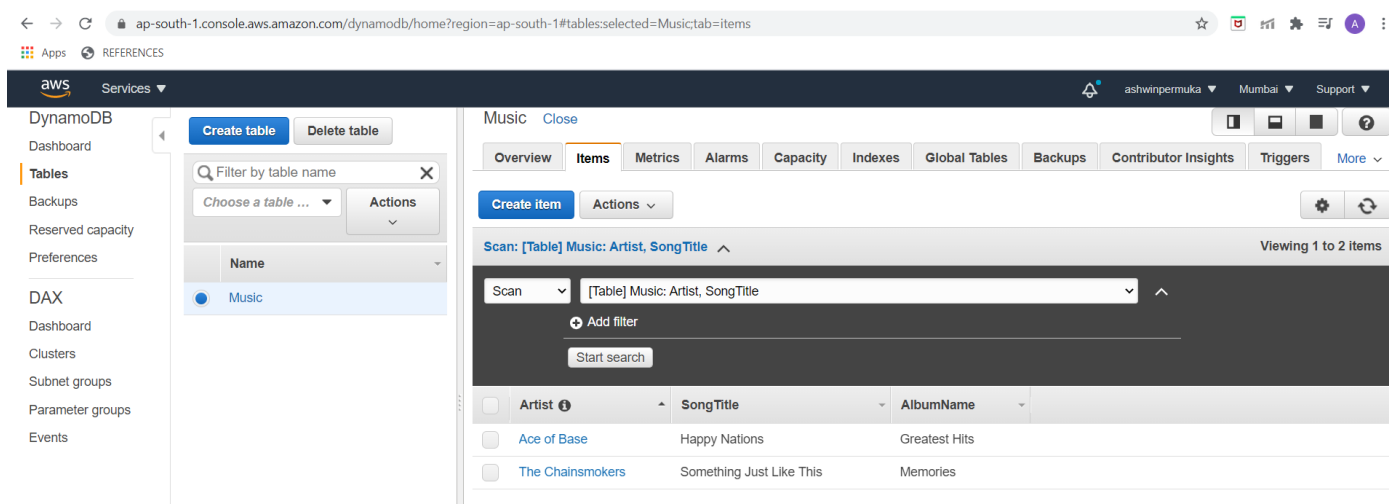
```
Command Prompt
C:\Users\ashwi>aws dynamodb put-item --table-name Music --item file://item_insert1.json

C:\Users\ashwi>aws dynamodb describe-table --table-name Music
{
  "Table": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "Music",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "ACTIVE",
    "CreationDateTime": "2020-11-12T12:25:10.497000+05:30",
```

```
Command Prompt
C:\Users\ashwi>aws dynamodb put-item --table-name Music --item file:///item_insert2.json

C:\Users\ashwi>aws dynamodb scan --table-name Music
{
  "Items": [
    {
      "Artist": {
        "S": "Ace of Base"
      },
      "AlbumName": {
        "S": "Greatest Hits"
      },
      "SongTitle": {
        "S": "Happy Nations"
      }
    },
    {
      "Artist": {
        "S": "The Chainsmokers"
      },
      "AlbumName": {
        "S": "Memories"
      },
      "SongTitle": {
        "S": "Something Just Like This"
      }
    }
  ],
  "Count": 2,
  "ScannedCount": 2,
  "ConsumedCapacity": null
}
```

After inserting those 2 items:-



Scan the table to see all items:

```
Command Prompt
C:\Users\ashwi>aws dynamodb put-item --table-name Music --item file:///item_insert2.json

C:\Users\ashwi>aws dynamodb scan --table-name Music
{
  "Items": [
    {
      "Artist": {
        "S": "Ace of Base"
      },
      "AlbumName": {
        "S": "Greatest Hits"
      },
      "SongTitle": {
        "S": "Happy Nations"
      }
    },
    {
      "Artist": {
        "S": "The Chainsmokers"
      },
      "AlbumName": {
        "S": "Memories"
      },
      "SongTitle": {
        "S": "Something Just Like This"
      }
    }
  ],
  "Count": 2,
  "ScannedCount": 2,
  "ConsumedCapacity": null
}
```

Query to project “SongTitle” of all items where Artist is “The Chainsmokers” :-

```
Command Prompt
C:\Users\ashwi>aws dynamodb query --table-name Music --projection-expression "SongTitle" --key-condition-expression "Artist = :v1" --expression-attribute-values file://query1.json --return-consumed-capacity TOTAL
{
  "Items": [
    {
      "SongTitle": {
        "S": "Something Just Like This"
      }
    }
  ],
  "Count": 1,
  "ScannedCount": 1,
  "ConsumedCapacity": {
    "TableName": "Music",
    "CapacityUnits": 0.5
  }
}
```

query1.json:

```
query1 - Notepad
File Edit Format View Help
{
  ":v1": {"S": "The Chainsmokers"}
}
```

Inserting one more item:-

```
Command Prompt
C:\Users\ashwi>aws dynamodb put-item --table-name Music --item file://item3.json

C:\Users\ashwi>aws dynamodb scan --table-name Music
{
  "Items": [
    {
      "Artist": {
        "S": "BTS"
      },
      "Genre": {
        "S": "Mood Booster"
      },
      "AlbumName": {
        "S": "Dynamite Melodies"
      },
      "SongTitle": {
        "S": "Dynamite"
      }
    },
    {
      "Artist": {
        "S": "Ace of Base"
      },
      "AlbumName": {
        "S": "Greatest Hits"
      },
      "SongTitle": {
        "S": "Happy Nations"
      }
    },
    {
      "Artist": {
        "S": "The Chainsmokers"
      },
      "AlbumName": {
        "S": "Memories"
      },
      "SongTitle": {
        "S": "Something Just Like This"
      }
    }
  ],
  "Count": 3,
  "ScannedCount": 3,
  "ConsumedCapacity": null
}
```

item3.json

```
item3 - Notepad
File Edit Format View Help
{
  "Artist": {"S": "BTS"},
  "SongTitle": {"S": "Dynamite"},
  "AlbumName": {"S": "Dynamite Melodies"},
  "Genre": {"S": "Mood Booster"}
}
```

After inserting item3:

The screenshot shows the AWS Management Console for a DynamoDB instance. The left sidebar shows the 'DynamoDB' service with a 'Tables' section. The 'Music' table is selected. The 'Items' tab is active, showing a list of items. The table has a primary key 'Artist' and a secondary index 'SongTitle'. The items are:

| Artist | SongTitle | AlbumName | Genre |
|------------------|--------------------------|-------------------|--------------|
| Ace of Base | Happy Nations | Greatest Hits | |
| BTS | Dynamite | Dynamite Melodies | Mood Booster |
| The Chainsmokers | Something Just Like This | Memories | |

Updating the table :-

Updating the provisioned-throughput ReadCapacity and WriteCapacity units:

```
Command Prompt
C:\Users\ashwi>aws dynamodb update-table --table-name Music --provisioned-throughput ReadCapacityUnits=20,WriteCapacityUnits=10
{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "Music",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "UPDATING",
    "CreationDateTime": "2020-11-12T12:25:10.497000+05:30",
    "ProvisionedThroughput": {
      "LastIncreaseDateTime": "2020-11-12T13:29:50.877000+05:30",
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:ap-south-1:207477500461:table/Music",
    "TableId": "e32e141f-69ea-44b4-a870-44daf4ecb056"
  }
}
```

```
C:\Users\ashwi>aws dynamodb update-table --table-name Music --provisioned-throughput ReadCapacityUnits=8,WriteCapacityUnits=4
{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "Music",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "UPDATING",
    "CreationDateTime": "2020-11-12T12:25:10.497000+05:30",
    "ProvisionedThroughput": {
      "LastIncreaseDateTime": "2020-11-12T13:29:53.988000+05:30",
      "LastDecreaseDateTime": "2020-11-12T13:31:02.912000+05:30",
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 20,
      "WriteCapacityUnits": 10
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:ap-south-1:207477500461:table/Music",
    "TableId": "e32e141f-69ea-44b4-a870-44daf4ecb056"
  }
}
```

Updating the table by adding new attribute called “Price”:

```
Command Prompt
C:\Users\ashwi>aws dynamodb put-item --table-name Music --item file://update_price.json

C:\Users\ashwi>aws dynamodb scan --table-name Music
{
  "Items": [
    {
      "Price": {
        "N": "3.5"
      },
      "Artist": {
        "S": "BTS"
      },
      "Genre": {
        "S": "Mood Booster"
      },
      "AlbumName": {
        "S": "Dynamite Melodies"
      },
      "SongTitle": {
        "S": "Dynamite"
      }
    },
    {
      "Artist": {
        "S": "Ace of Base"
      },
      "AlbumName": {
        "S": "Greatest Hits"
      },
      "SongTitle": {
        "S": "Happy Nations"
      }
    },
    {
      "Artist": {
        "S": "The Chainsmokers"
      },
      "AlbumName": {
        "S": "Memories"
      },
      "SongTitle": {
        "S": "Something Just Like This"
      }
    }
  ],
  "Count": 3,
  "ScannedCount": 3,
  "ConsumedCapacity": null
}
```

update_price.json

```
update_price - Notepad
File Edit Format View Help
{
  "Artist": {"S": "BTS"},
  "SongTitle": {"S": "Dynamite"},
  "AlbumName": {"S": "Dynamite Melodies"},
  "Genre": {"S": "Mood Booster"},
  "Price": {"N": "3.5"}
}
```

After the updating the table by adding Price:

The screenshot shows the AWS DynamoDB console interface. On the left, the 'DynamoDB' sidebar is visible with options like 'Tables', 'Backups', and 'Reserved capacity'. The main panel displays the 'Music' table. The 'Items' tab is selected, showing a list of items with columns: Artist, SongTitle, AlbumName, Genre, and Price. The items are: Ace of Base (Happy Nations, Greatest Hits), BTS (Dynamite, Dynamite Melodies, Mood Booster, 3.5), and The Chainsmokers (Something Just Like This, Memories). The 'Price' column is highlighted in blue, indicating it is the selected column for the current view.

| Artist | SongTitle | AlbumName | Genre | Price |
|------------------|--------------------------|-------------------|--------------|-------|
| Ace of Base | Happy Nations | Greatest Hits | | |
| BTS | Dynamite | Dynamite Melodies | Mood Booster | 3.5 |
| The Chainsmokers | Something Just Like This | Memories | | |

Updating the item by changing the value of “Price” of a particular item:

```
Command Prompt
C:\Users\ashwi>aws dynamodb update-item --table-name Music --key file://key1.json --update-expression "SET #P = :P" --expression-attribute-names file://u1.json --expression-attribute-values file://u1_values.json
--return-values ALL_NEW
{
  "Attributes": {
    "Price": {
      "N": "5.25"
    },
    "AlbumName": {
      "S": "Dynamite Melodies"
    },
    "Artist": {
      "S": "BTS"
    },
    "Genre": {
      "S": "Mood Booster"
    },
    "SongTitle": {
      "S": "Dynamite"
    }
  }
}
```

u1.json and u1_values.json

| u1 - Notepad | u1_values - Notepad |
|--------------------------------|--------------------------------------|
| <pre>{ "#P": "Price" }</pre> | <pre>{ ":P": {"N": "5.25"} }</pre> |

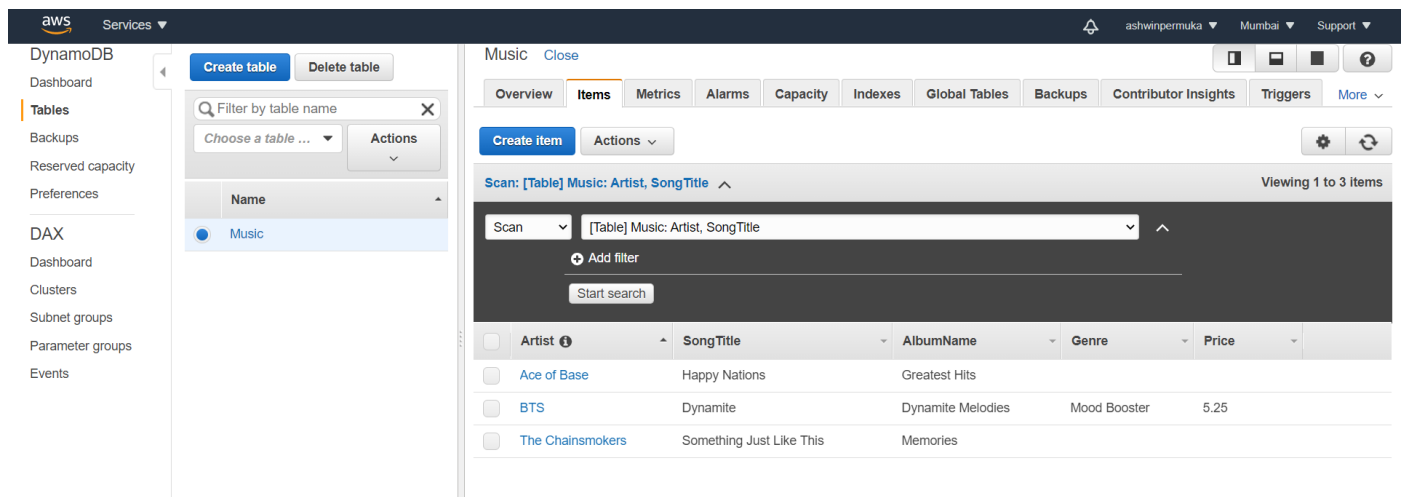
key1.json

```
key1 - Notepad
File Edit Format View Help
{
  "Artist": {"S": "BTS"},
  "SongTitle": {"S": "Dynamite"}
}
```

Scanning entire table after the update:

```
Command Prompt
C:\Users\ashwi>aws dynamodb scan --table-name Music
{
  "Items": [
    {
      "Price": {
        "N": "5.25"
      },
      "AlbumName": {
        "S": "Dynamite Melodies"
      },
      "Artist": {
        "S": "BTS"
      },
      "Genre": {
        "S": "Mood Booster"
      },
      "SongTitle": {
        "S": "Dynamite"
      }
    },
    {
      "Artist": {
        "S": "Ace of Base"
      },
      "AlbumName": {
        "S": "Greatest Hits"
      },
      "SongTitle": {
        "S": "Happy Nations"
      }
    },
    {
      "Artist": {
        "S": "The Chainsmokers"
      },
      "AlbumName": {
        "S": "Memories"
      },
      "SongTitle": {
        "S": "Something Just Like This"
      }
    }
  ],
  "Count": 3,
  "ScannedCount": 3,
  "ConsumedCapacity": null
}
```

We can observe the same change in aws console also:



The screenshot shows the AWS Management Console for a DynamoDB table named 'Music'. The 'Items' tab is active, showing a list of items. The table has a primary key 'Artist' and a global secondary index 'AlbumName_Index'. The items listed are:

| Artist | SongTitle | AlbumName | Genre | Price |
|------------------|--------------------------|-------------------|--------------|-------|
| Ace of Base | Happy Nations | Greatest Hits | | |
| BTS | Dynamite | Dynamite Melodies | Mood Booster | 5.25 |
| The Chainsmokers | Something Just Like This | Memories | | |

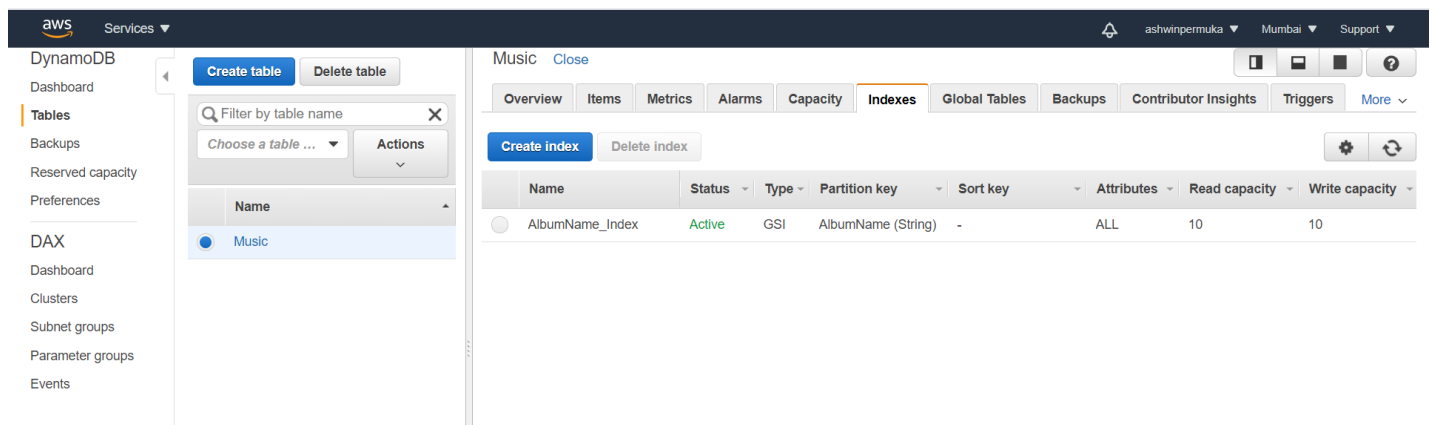
Updating the table by creating a global secondary index:

```
C:\Users\ashwi>aws dynamodb update-table --table-name Music --attribute-definitions AttributeName=AlbumName,AttributeType=S --global-secondary-index-updates file://gsi.json
{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "AlbumName",
        "AttributeType": "S"
      },
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "Music",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "UPDATING",
    "CreationDateTime": "2020-11-12T12:25:10.497000+05:30",
    "ProvisionedThroughput": {
      "LastIncreaseDateTime": "2020-11-12T13:31:25.647000+05:30",
      "LastDecreaseDateTime": "2020-11-12T13:31:06.916000+05:30",
      "NumberOfDecreasesToday": 1,
      "ReadCapacityUnits": 20,
      "WriteCapacityUnits": 10
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:ap-south-1:207477500461:table/Music",
    "TableId": "c32e141f-69ea-44b4-a070-44da74ecb050",
    "GlobalSecondaryIndexes": [
      {
        "IndexName": "AlbumName_Index",
        "KeySchema": [
          {
            "AttributeName": "AlbumName",
            "KeyType": "HASH"
          }
        ],
        "Projection": {
          "ProjectionType": "ALL"
        },
        "IndexStatus": "CREATING",
        "Backfilling": false,
        "ProvisionedThroughput": {
          "NumberOfDecreasesToday": 0,
          "ReadCapacityUnits": 10,
          "WriteCapacityUnits": 10
        }
      }
    ]
  }
}
```

Gsi.json file:

```
gsi - Notepad
File Edit Format View Help
[
  {
    "Create": {
      "IndexName": "AlbumName_Index",
      "KeySchema": [
        {
          "AttributeName": "AlbumName",
          "KeyType": "HASH"
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 10,
        "WriteCapacityUnits": 10
      },
      "Projection": {
        "ProjectionType": "ALL"
      }
    }
  }
]
```


After creating the index:



Deleting an item from table:-

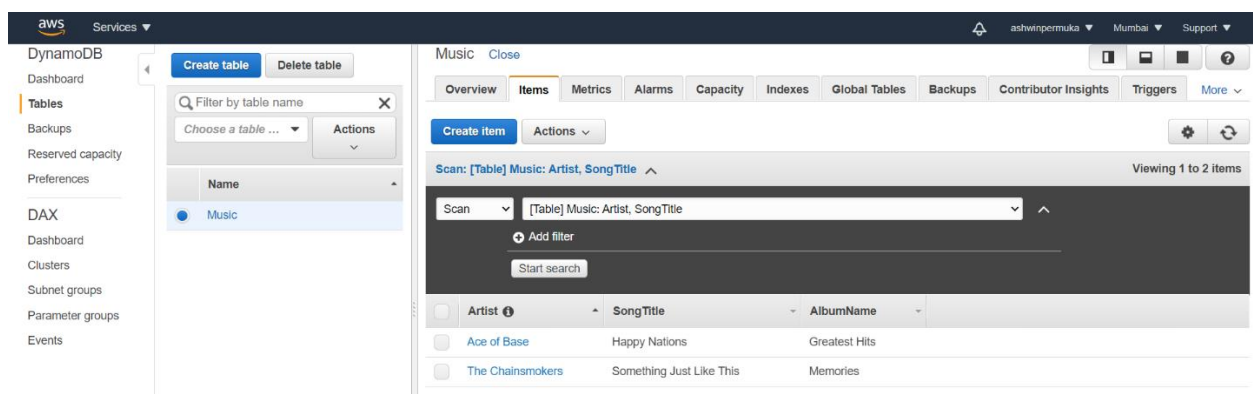
To delete an item from Music table, specifying delete item as the key,

```
Command Prompt
C:\Users\ashwi>aws dynamodb delete-item --table-name Music --key file://key1.json --return-consumed-capacity TOTAL
{
  "ConsumedCapacity": {
    "TableName": "Music",
    "CapacityUnits": 2.0
  }
}

C:\Users\ashwi>aws dynamodb scan --table-name Music
{
  "Items": [
    {
      "Artist": {
        "S": "Ace of Base"
      },
      "AlbumName": {
        "S": "Greatest Hits"
      },
      "SongTitle": {
        "S": "Happy Nations"
      }
    },
    {
      "Artist": {
        "S": "The Chainsmokers"
      },
      "AlbumName": {
        "S": "Memories"
      },
      "SongTitle": {
        "S": "Something Just Like This"
      }
    }
  ],
  "Count": 2,
  "ScannedCount": 2,
  "ConsumedCapacity": null
}
```

```
key1 - Notepad
File Edit Format View Help
{
  "Artist": {"S": "BTS"},
  "SongTitle": {"S": "Dynamite"}
}
```

After deleting the item, there are only 2 items now:



Deleting a table:-

Here I create a new table called 'NewMusic' and then I delete it.

Creating the table NewMusic:

```
Command Prompt
C:\Users\ashwi>aws dynamodb list-tables
{
  "TableNames": [
    "Music"
  ]
}

C:\Users\ashwi>aws dynamodb create-table --table-name NewMusic --attribute-definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S --key-schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5
{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "NewMusic",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "CREATING",
    "CreationDateTime": "2020-11-12T14:45:55.772000+05:30",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:ap-south-1:207477500461:table/NewMusic",
    "TableId": "71116568-7dc1-4dba-897c-558c69d7b5f8"
  }
}

C:\Users\ashwi>aws dynamodb list-tables
{
  "TableNames": [
    "Music",
    "NewMusic"
  ]
}
```

Now we have 2 tables Music and NewMusic:

The screenshot shows the AWS Management Console for DynamoDB. The 'Tables' tab is selected, displaying a list of two tables: 'Music' and 'NewMusic'. Both tables are in an 'Active' state. The 'Music' table has a primary key of 'Artist (String)' and a sort key of 'SongTitle (String)', with 1 index, 30 read capacity units, and 20 write capacity units. The 'NewMusic' table has the same primary and sort keys but with 0 indices, 10 read capacity units, and 5 write capacity units.

| Name | Status | Partition key | Sort key | Indexes | Total read capacity | Total write capacity |
|----------|--------|-----------------|--------------------|---------|---------------------|----------------------|
| Music | Active | Artist (String) | SongTitle (String) | 1 | 30 | 20 |
| NewMusic | Active | Artist (String) | SongTitle (String) | 0 | 10 | 5 |

Deleting the table NewMusic:-

We use delete-table command to delete a table

```
Command Prompt
C:\Users\ashwi>aws dynamodb delete-table --table-name NewMusic
{
  "TableDescription": {
    "TableName": "NewMusic",
    "TableStatus": "DELETING",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:ap-south-1:207477500461:table/NewMusic",
    "TableId": "71116568-7dc1-4dba-897c-558c69d7b5f8"
  }
}

C:\Users\ashwi>aws dynamodb list-tables
{
  "TableNames": [
    "Music"
  ]
}

C:\Users\ashwi>
```

Now we can see that the NewMusic table has been successfully deleted:

aws

Services

ashwinpermuka

Mumbai

Support

DynamoDB

Dashboard

Tables

Backups

Reserved capacity

Preferences

DAX

Dashboard

Clusters

Subnet groups

Parameter groups

Events

Create table

Delete table

Filter by table name

Choose a table group

Actions

1 to 1 of 1 Tables

| Name | Status | Partition key | Sort key | Indexes | Total read capacity | Total write capacity |
|-------|--------|-----------------|--------------------|---------|---------------------|----------------------|
| Music | Active | Artist (String) | SongTitle (String) | 1 | 30 | 20 |

Thank you