

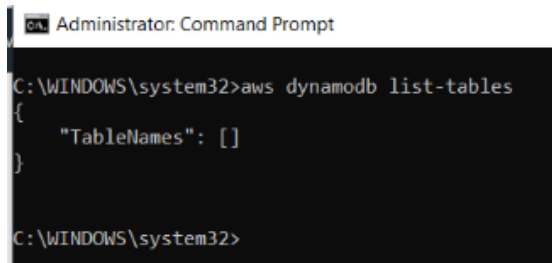
DBT ASSIGNMENT 4

Sneha Hegde

PES1201801157

2. DynamoDB-

- Listing all the tables initially gives an empty list-



```
Administrator: Command Prompt
C:\WINDOWS\system32>aws dynamodb list-tables
{
  "TableNames": []
}
C:\WINDOWS\system32>
```

- Creating a table, Musician, and adding two attributes, musician_id (PK) and stage_name (sort key), along with their types-

```
aws dynamodb create-table --table-name Musician --attribute-definitions
AttributeName=musician_id,AttributeType=S
AttributeName=stage_name,AttributeType=S --key-schema
AttributeName=musician_id,KeyType=HASH
AttributeName=stage_name,KeyType=RANGE --provisioned-throughput
ReadCapacityUnits=1,WriteCapacityUnits=1
```

```

C:\WINDOWS\system32>aws dynamodb create-table --table-name Musician --attribute-definitions AttributeName=musician_id,AttributeType=S AttributeName=stage_name,AttributeType=S --key-schema AttributeName=musician_id,KeyType=HASH AttributeName=stage_name,KeyType=RANGE --provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnits=1
{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "musician_id",
        "AttributeType": "S"
      },
      {
        "AttributeName": "stage_name",
        "AttributeType": "S"
      }
    ],
    "TableName": "Musician",
    "KeySchema": [
      {
        "AttributeName": "musician_id",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "stage_name",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "CREATING",
    "CreationDateTime": "2020-11-05T21:15:40.332000+05:30",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 1,
      "WriteCapacityUnits": 1
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-east-1:249484679381:table/Musician",
    "TableId": "5b2ccf78-5bf5-48d3-95ef-5c1cc79b12c9"
  }
}
C:\WINDOWS\system32>

```

- Listing the tables again, the newly created Musician table is seen-

```

C:\WINDOWS\system32>aws dynamodb list-tables
{
  "TableNames": [
    "Musician"
  ]
}

C:\WINDOWS\system32>_

```

- Insertion into Musician using put command-

aws dynamodb put-item --table-name Musician --item file://insert.json

aws dynamodb put-item --table-name Musician --item file://insert2.json

```
D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>aws dynamodb put-item --table-name Musician --item file://insert.json
D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>
```

Both the items
have been
inserted.

```
D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>aws dynamodb scan --table-name Musician
{
  "Items": [
    {
      "musician_id": {
        "S": "mus1"
      },
      "no_of_albums": {
        "N": "6"
      },
      "best_songs": {
        "SS": [
          "song1",
          "song2"
        ]
      },
      "stage_name": {
        "S": "Roxy"
      },
      "age": {
        "N": "20"
      }
    },
    {
      "awards": {
        "S": "musicAward"
      },
      "musician_id": {
        "S": "mus2"
      },
      "no_of_albums": {
        "N": "8"
      },
      "best_songs": {
        "SS": [
          "song3",
          "song4"
        ]
      },
      "stage_name": {
        "S": "Trixie"
      },
      "age": {
        "N": "35"
      }
    }
  ]
}
-- More --
```

- Retrieve the stage name of the item where musician id is mus1-

aws dynamodb query --table-name Musician --projection-expression "stage_name" --key-condition-expression "musician_id = :v1" --expression-attribute-values file://query1.json --return-consumed-capacity TOTAL

```
D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>aws dynamodb query --table-name Musician --projection-expression "stage_name" --key-condition-expression "musician_id = :v1" --expression-attribute-values file://query1.json --return-consumed-capacity TOTAL
{
  "Items": [
    {
      "stage_name": {
        "S": "Roxy"
      }
    }
  ],
  "Count": 1,
  "ScannedCount": 1,
  "ConsumedCapacity": {
    "TableName": "Musician",
    "CapacityUnits": 0.5
  }
}

D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>
```

- Update using put command-

Updating the first item to include an award attribute-

aws dynamodb put-item --table-name Musician --item file://update1.json

It has been updated, the new field has been added to item 1.

```
D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>aws dynamodb put-item --table-name Musician --item file://update1.json
D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>aws dynamodb scan --table-name Musician
{
  "Items": [
    {
      "awards": {
        "S": "AWARD"
      },
      "musician_id": {
        "S": "mus1"
      },
      "no_of_albums": {
        "N": "6"
      },
      "best_songs": {
        "SS": [
          "song1",
          "song2"
        ]
      },
      "stage_name": {
        "S": "Roxy"
      },
      "age": {
        "N": "20"
      }
    },
    {
      "awards": {
        "S": "musicAward"
      },
      "musician_id": {
        "S": "mus2"
      },
      "no_of_albums": {
        "N": "8"
      },
      "best_songs": {
        "SS": [
          "song3",
          "song4"
        ]
      },
      "stage_name": {
        "S": "Trixie"
      }
    }
  ]
}
```

- Update using update command-
Update Trixie's number of albums to 10-

aws dynamodb update-item --table-name Musician --key file://key.json --update-expression "SET #V = :v" --expression-attribute-names file://u1.json --expression-attribute-values file://u2.json --return-values ALL_NEW

```
D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>aws dynamodb update-item --table-name Musician --key file://key.json --update-expression "SET #V = :v" --expression-attribute-names file://u1.json --expression-attribute-values file://u2.json --return-values ALL_NEW
{
  "Attributes": {
    "awards": {
      "S": "musicAward"
    },
    "musician_id": {
      "S": "mus2"
    },
    "no_of_albums": {
      "N": "10"
    },
    "best_songs": {
      "SS": [
        "song3",
        "song4"
      ]
    },
    "stage_name": {
      "S": "Trixie"
    },
    "age": {
      "N": "35"
    }
  }
}
```

D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>_

It has been updated.

- Delete item where name is Roxy-

aws dynamodb delete-item
--table-name Musician --key
file://delete.json --return-
consumed-capacity TOTAL

It has been deleted.

```
D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>aws dynamodb delete-item --table-name Musician --key file://delete.json --return-consumed-capacity TOTAL
{
  "ConsumedCapacity": {
    "TableName": "Musician",
    "CapacityUnits": 1.0
  }
}
```

```
D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>aws dynamodb scan --table-name Musician
{
  "Items": [
    {
      "awards": {
        "S": "musicAward"
      },
      "musician_id": {
        "S": "mus2"
      },
      "no_of_albums": {
        "N": "10"
      },
      "best_songs": {
        "SS": [
          "song3",
          "song4"
        ]
      },
      "stage_name": {
        "S": "Trixie"
      },
      "age": {
        "N": "35"
      }
    }
  ],
  "Count": 1,
  "ScannedCount": 1,
  "ConsumedCapacity": null
}
```

D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>_

- delete the table Musician-

aws dynamodb delete-table --table-name Musician

```
D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>aws dynamodb delete-table --table-name Musician
{
  "TableDescription": {
    "TableName": "Musician",
    "TableStatus": "DELETING",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 1,
      "WriteCapacityUnits": 1
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-east-1:249484679381:table/Musician",
    "TableId": "5b2ccf78-5bf5-48d3-95ef-5c1cc79b12c9"
  }
}

D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>aws dynamodb list-tables
{
  "TableNames": []
}

D:\PES_Stuff\SEM_5_STUFF\DBT\Assignments\Assignment4>_
```

The table has been deleted and hence, the list is empty.