

CSCI 5410: Assignment 1

Part C

1. List of Super Volcanoes:

Table 1. Super Volcanoes [1][2]

Name	Place/ Location	Properties	Last VEI	Last Major Eruption	Size
Long Valley Caldera	East Central California	Lat: 37.72 Long: -118.88	7.00	760,000 years ago	200-square-mile
Valles Caldera	New Mexico	Lat: 35.90 Long: -106.53	7.00	1.2 million and 1.6 million years ago	175-square-mile
Lake Toba	North Sumatra, Indonesia	Lat: 2.79 Long: 98.62	8.00	75,000 years ago	1,080-square-mile
Aira Caldera	Kagoshima, Japan	Lat: 31.59 Long: 130.66	7.00	22,000 years ago	150-square-mile
Taupo Caldera	New Zealand	Lat: -38.80 Long: 175.90	8.00	26,500 years ago	485-square-mile
Yellowstone Caldera	Yellowstone National Park	Lat: 44.41 Long: -110.72	8.00	640,000 years ago	30 by 45 miles

2. Screenshots of DynamoDB:

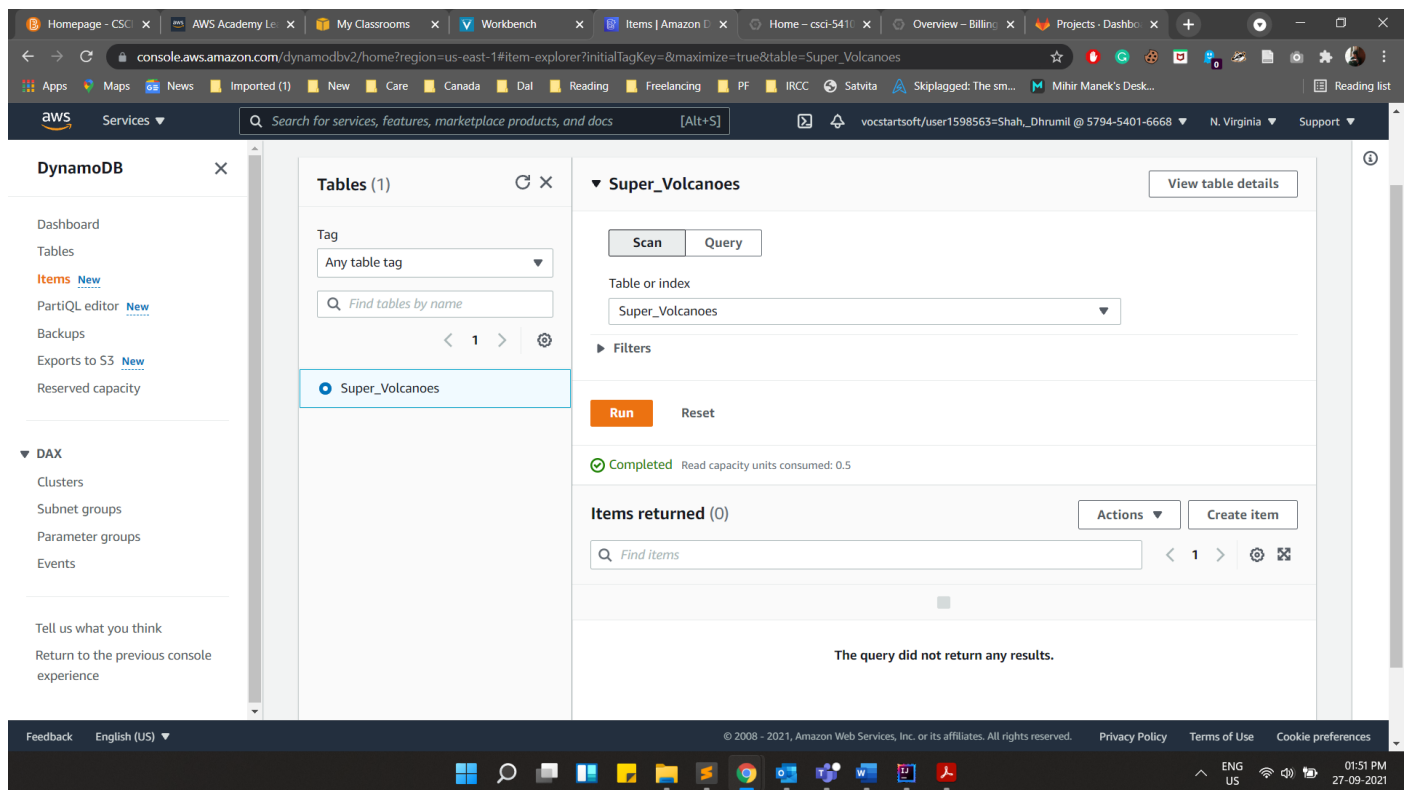


Figure 1: AWS Empty DynamoDB

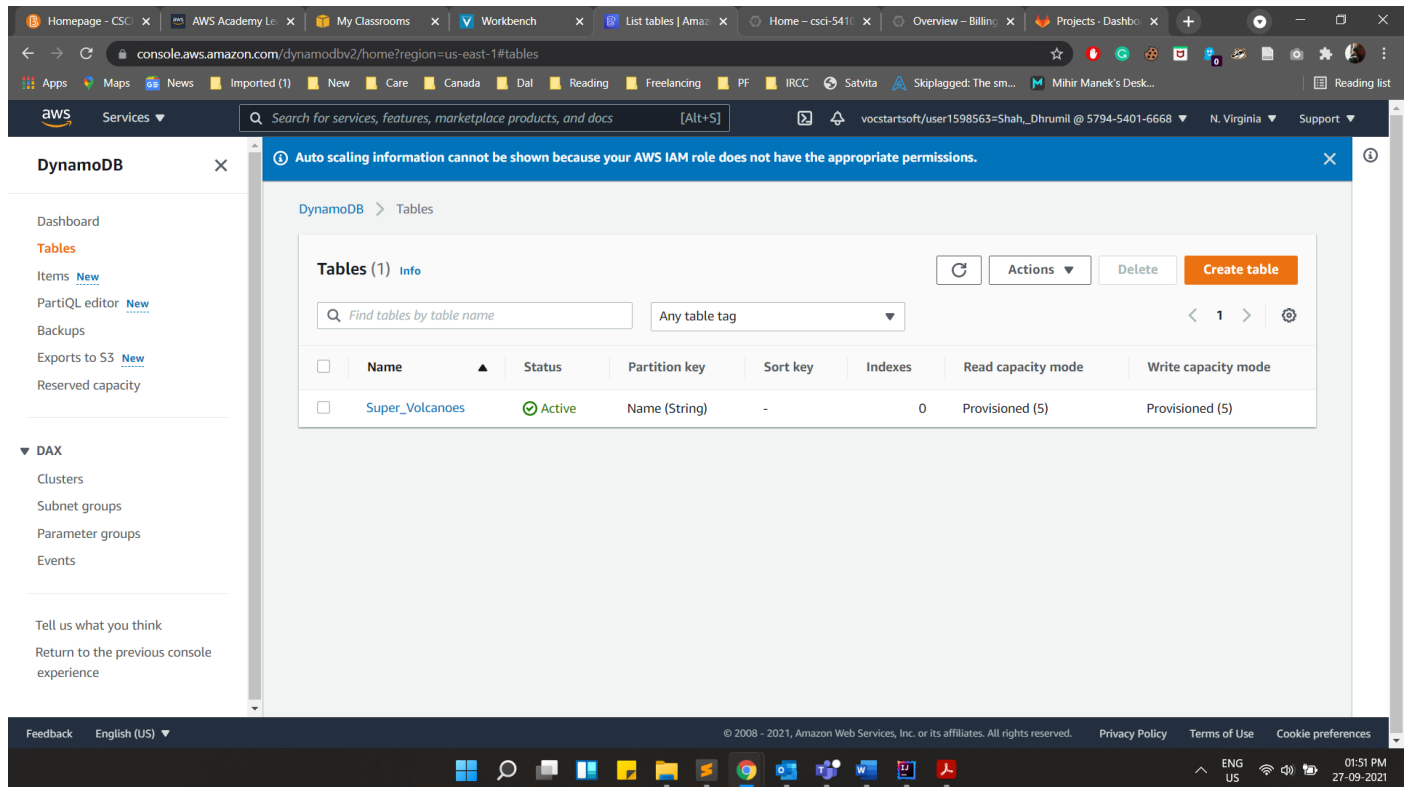


Figure 2: AWS DynamoDB Super_Volcanoes Table

The screenshot shows the AWS DynamoDB console interface. The left sidebar contains navigation links for Dashboard, Tables, Items (highlighted), PartiQL editor, Backups, Exports to S3, and Reserved capacity. Under the DAX section, there are links for Clusters, Subnet groups, Parameter groups, and Events. The main content area displays the 'Super_Volcanoes' table. A 'Run' button is visible, and a status message indicates 'Completed' with 'Read capacity units consumed: 0.5'. Below this, a section titled 'Items returned (6)' shows a list of six items. Each item has a checkbox, a name, a 'Last_VEI' value, a location, properties (latitude and longitude), and a size. The items are: Long Valley Caldera, Valles Caldera, Taupo Caldera, Aira Caldera, Lake Toba, and Yellowstone Caldera.

	Name	Last_VEI	Location	Properties	Size
<input type="checkbox"/>	Long Valley Caldera	7	East Central California	Lat: 37.72 Long: -118.88	200-square-mile
<input type="checkbox"/>	Valles Caldera	7	New Mexico	Lat: 35.90 Long: -106.53	175-square-mile
<input type="checkbox"/>	Taupo Caldera	8	New Zealand	Lat: -38.80 Long: 175.90	485-square-mile
<input type="checkbox"/>	Aira Caldera	7	Kagoshima, Japan	Lat: 31.59 Long: 130.66	150-square-mile
<input type="checkbox"/>	Lake Toba	8	North Sumatra, Indonesia	Lat: 2.79 Long: 98.62	1,080-square-mile
<input type="checkbox"/>	Yellowstone Caldera	8	Yellowstone National Park	Lat: 44.41 Long: -110.72	30 by 45 miles

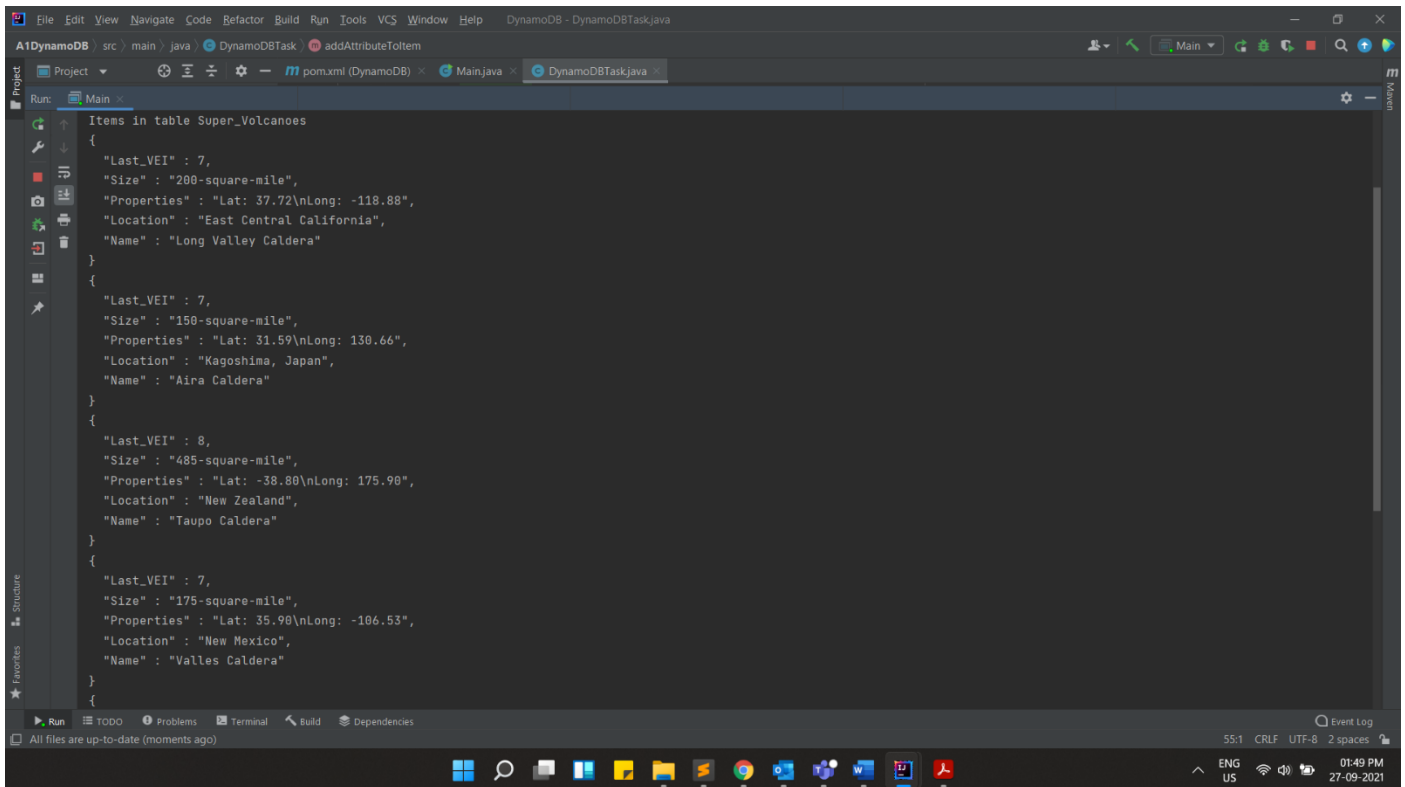
Figure 3: AWS Data Inserted DynamoDB

This screenshot shows the same AWS DynamoDB console interface as Figure 3, but with updated data. The 'Items returned (6)' section now includes an additional column, 'last_eruption_period', which provides the time elapsed since the last eruption for each volcano. The status message remains 'Completed' with 'Read capacity units consumed: 0.5'.

	Name	last_eruption_period	Last_VEI	Location	Properties	Size
<input type="checkbox"/>	Long Valley...	760,000 years ago	7	East Centra...	Lat: 37.72 L...	200-square-mile
<input type="checkbox"/>	Valles Caldera	1.2 million and 1.6 million years ago	7	New Mexico	Lat: 35.90 L...	175-square-mile
<input type="checkbox"/>	Taupo Cald...	26,500 years ago	8	New Zealand	Lat: -38.80 ...	485-square-mile
<input type="checkbox"/>	Aira Caldera	22,000 years ago	7	Kagoshima,...	Lat: 31.59 L...	150-square-mile
<input type="checkbox"/>	Lake Toba	75,000 years ago	8	North Sum...	Lat: 2.79 Lo...	1,080-square-mile
<input type="checkbox"/>	Yellowston...	640,000 years ago	8	Yellowston...	Lat: 44.41 L...	30 by 45 miles

Figure 4: AWS Updated DynamoDB

3. Code Output of DynamoDB:

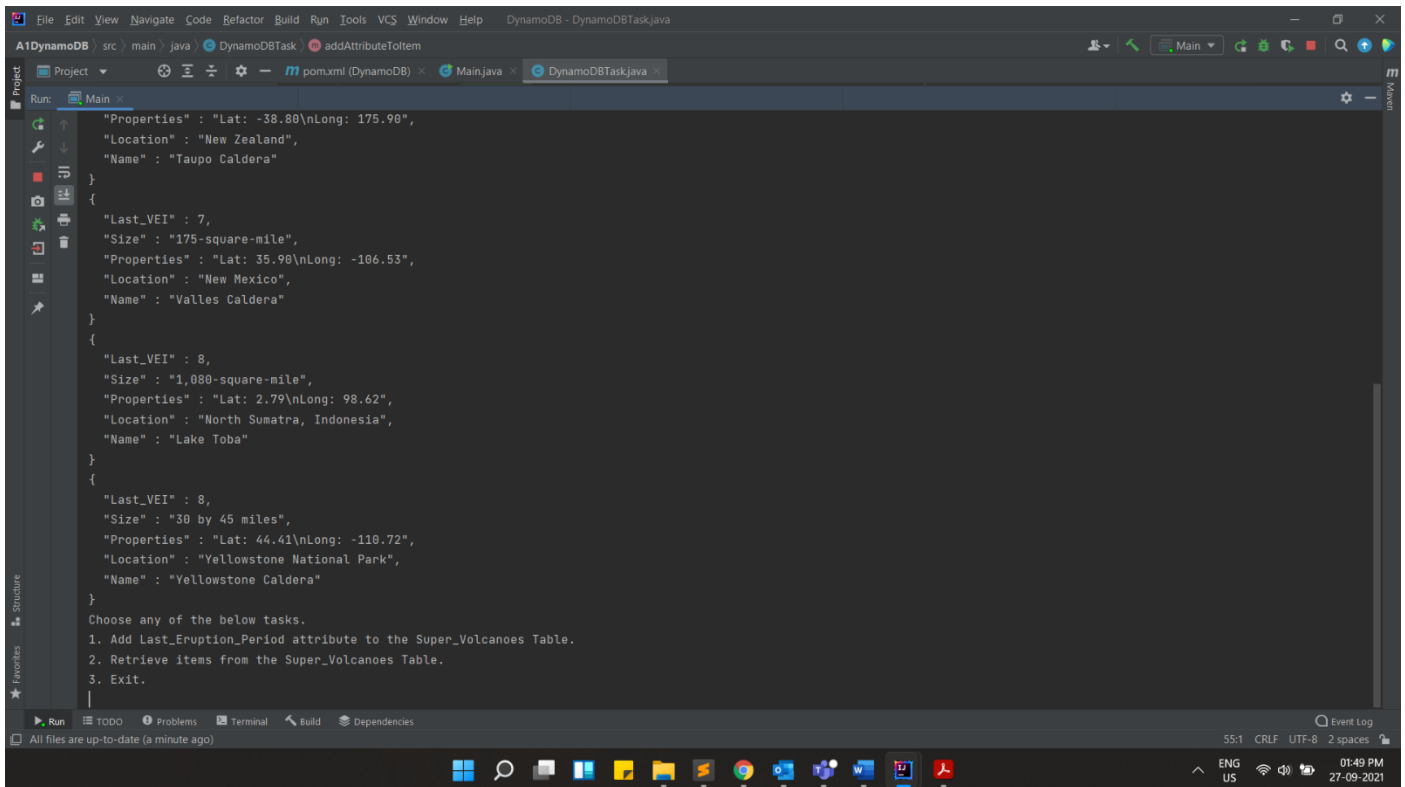


```

Run: Main
Items in table Super_Volcanoes
{
  "Last_VEI" : 7,
  "Size" : "200-square-mile",
  "Properties" : "Lat: 37.72\nLong: -118.88",
  "Location" : "East Central California",
  "Name" : "Long Valley Caldera"
}
{
  "Last_VEI" : 7,
  "Size" : "150-square-mile",
  "Properties" : "Lat: 31.59\nLong: 130.66",
  "Location" : "Kagoshima, Japan",
  "Name" : "Aira Caldera"
}
{
  "Last_VEI" : 8,
  "Size" : "485-square-mile",
  "Properties" : "Lat: -38.80\nLong: 175.98",
  "Location" : "New Zealand",
  "Name" : "Taupo Caldera"
}
{
  "Last_VEI" : 7,
  "Size" : "175-square-mile",
  "Properties" : "Lat: 35.98\nLong: -106.53",
  "Location" : "New Mexico",
  "Name" : "Valles Caldera"
}

```

Figure 5: Code Output Inserted Data DynamoDB 1



```

Run: Main
"Properties" : "Lat: -38.80\nLong: 175.98",
"Location" : "New Zealand",
"Name" : "Taupo Caldera"
}
{
  "Last_VEI" : 7,
  "Size" : "175-square-mile",
  "Properties" : "Lat: 35.98\nLong: -106.53",
  "Location" : "New Mexico",
  "Name" : "Valles Caldera"
}
{
  "Last_VEI" : 8,
  "Size" : "1,080-square-mile",
  "Properties" : "Lat: 2.79\nLong: 98.62",
  "Location" : "North Sumatra, Indonesia",
  "Name" : "Lake Toba"
}
{
  "Last_VEI" : 8,
  "Size" : "30 by 45 miles",
  "Properties" : "Lat: 44.41\nLong: -110.72",
  "Location" : "Yellowstone National Park",
  "Name" : "Yellowstone Caldera"
}
Choose any of the below tasks.
1. Add Last_Eruption_Period attribute to the Super_Volcanoes Table.
2. Retrieve items from the Super_Volcanoes Table.
3. Exit.

```

Figure 6: Code Output Inserted Data DynamoDB 2

```
Choose any of the below tasks.
1. Add Last_Eruption_Period attribute to the Super_Volcanoes Table.
2. Retrieve items from the Super_Volcanoes Table.
3. Exit.

Items in table Super_Volcanoes
{
  "Last_VEI" : 7,
  "Last_eruption_period" : "760,000 years ago",
  "Size" : "200-square-mile",
  "Properties" : "Lat: 37.72\nLong: -118.88",
  "Location" : "East Central California",
  "Name" : "Long Valley Caldera"
}
{
  "Last_VEI" : 7,
  "Last_eruption_period" : "22,000 years ago",
  "Size" : "150-square-mile",
  "Properties" : "Lat: 31.59\nLong: 130.66",
  "Location" : "Kagoshima, Japan",
  "Name" : "Aira Caldera"
}
{
  "Last_VEI" : 8,
  "Last_eruption_period" : "26,500 years ago",
  "Size" : "485-square-mile",
  "Properties" : "Lat: -38.80\nLong: 175.90",
  "Location" : "New Zealand",
  "Name" : "Taupo Caldera"
}
```

Figure 7: Code Output Updated DynamoDB 1

```
Choose any of the below tasks.
1. Add Last_Eruption_Period attribute to the Super_Volcanoes Table.
2. Retrieve items from the Super_Volcanoes Table.
3. Exit.

{
  "Last_VEI" : 7,
  "Last_eruption_period" : "1.2 million and 1.6 million years ago",
  "Size" : "175-square-mile",
  "Properties" : "Lat: 35.98\nLong: -106.53",
  "Location" : "New Mexico",
  "Name" : "Valles Caldera"
}
{
  "Last_VEI" : 8,
  "Last_eruption_period" : "75,000 years ago",
  "Size" : "1,080-square-mile",
  "Properties" : "Lat: 2.79\nLong: 98.62",
  "Location" : "North Sumatra, Indonesia",
  "Name" : "Lake Toba"
}
{
  "Last_VEI" : 8,
  "Last_eruption_period" : "640,000 years ago",
  "Size" : "30 by 45 miles",
  "Properties" : "Lat: 44.41\nLong: -110.72",
  "Location" : "Yellowstone National Park",
  "Name" : "Yellowstone Caldera"
}
```

Figure 8: Code Output Updated DynamoDB 2

4. GitLab Repository Link:

https://git.cs.dal.ca/drshah/dhrumilrakeshshah_csci5410.git

5. Program Script:

Main.java:

```
import com.amazonaws.services.dynamodbv2.document.DynamoDB;

import java.util.Scanner;

/**
 * Author: Dhrumil Rakesh Shah
 * Version: 1.0
 * Class: The Main class containing the boilerplate code of AWS SDK for Java
 */
public class Main {

    // The driver method
    public static void main(String[] args) {

        // Instantiating the Scanner object to read user input
        Scanner sc = new Scanner(System.in);

        // Declaring & Initializing the choice to 0 that stores the
        // choice made by the user
        int choice = 0;

        // Storing the table name in a String
        String tableName = "Super_Volcanoes";

        // Creating a new DynamoDBTask class object
        DynamoDBTask dynamoDBTaskObject = new DynamoDBTask();

        // Calling the getConnection method in the DynamoDBTask class
        // Storing the connection object in dynamoDBObject
        DynamoDB dynamoDBObject = dynamoDBTaskObject.getConnection();

        // Looping through the menu
        while (choice != -1) {
            System.out.println("Choose any of the below tasks.");
            System.out.println("1. Add Last_Eruption_Period attribute to the Super_Volcanoes Table.");
            System.out.println("2. Retrieve items from the Super_Volcanoes Table.");
            System.out.println("3. Exit.");

            // Reading the user entered choice
            choice = sc.nextInt();

            // Switching through the different choices
            switch (choice) {
                case 1:
                    // Adding attribute to the Super_Volcanoes table
                    dynamoDBTaskObject.addAttributeToItem(tableName, dynamoDBObject);
                    break;
                case 2:
                    // Retrieving items form the Super_Volcanoes table
                    dynamoDBTaskObject.retrieveItem(tableName, dynamoDBObject);
            }
        }
    }
}
```

```

        break;
    case 3:
        // Exiting the application
        System.exit(0);
    default:
        // Default switch case
        System.out.println("Enter a valid option.");
        break;
    }
}
}
}
}

```

DynamoDBTask.java:

```

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDB;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;
import com.amazonaws.services.dynamodbv2.document.*;
import com.amazonaws.services.dynamodbv2.document.spec.UpdateItemSpec;
import com.amazonaws.services.dynamodbv2.document.utils.NameMap;
import com.amazonaws.services.dynamodbv2.document.utils.ValueMap;
import com.amazonaws.services.dynamodbv2.model.ReturnValue;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * Author: Dhrumil Rakesh Shah
 * Version: 1.0
 * Class: The AWS SDK for Java helper class
 */
public class DynamoDBTask {

    // The method to get connection object of the AWS account
    public DynamoDB getConnection() {

        // Storing the AWS credentials to establish the connection
        BasicSessionCredentials sessionCredentials = new BasicSessionCredentials(
            "ASIAYN2RGTSOE5Q4C5V4",
            "sRZ9yFppgxacwQKwD73vCMtwParngli8mffeRg6n",
            "FwoGZXIvYXZlEDYyADPkDdnVcKp2hbQrumCK9Afy4RrM7l6rVnVqRbkI" +
            "Gmy9ejYdcfUfPZA8lBAZIhT02zIZ9YkQYSNUOnmCYZUTHXBJjt9w1PdkNrXtK" +
            "0PXj0fcjOPsPLg8g91mJ+cgpb+N/McC+XL94Vl/pDkws0xEo1fZzUjZ4VLSXXA" +
            "z86jjU2LDpaSNZ10tPEwXhrx8dHU2BClyKaScAI6EfVltX+gd3JrcnWANwxB3C" +
            "QhmGna8Mxv4M1f2L4lE1Nq/kM7f3iSo/4jjsRqtitkKc537bBiix7caKBjItF38" +
            "tpoHmhJolOXAlqZEG9IBeg09ReYuyvU66KD7KR3ijp8rNDtuk82W4LQZ6");

        // Establishing connection with the Amazon DynamoDB service
        // and storing it in the amazonDynamoDB object
        AmazonDynamoDB amazonDynamoDB =
            AmazonDynamoDBClientBuilder.standard().withCredentials
                (new AWSStaticCredentialsProvider(sessionCredentials))
                .withRegion("us-east-1")

```

```

        .build();

/*
// Establishing connection with the Amazon DynamoDB service
// and storing it in the amazonDynamoDB
AmazonDynamoDB amazonDynamoDB = AmazonDynamoDBClientBuilder.standard()
    .withRegion(Regions.US_EAST_1)
    .build();
*/

// Instantiating DynamoDB by passing the connection object
DynamoDB dynamoDB = new DynamoDB(amazonDynamoDB);

// Printing if connection is successfully established
System.out.println("Connection Established.\n");

// Returning the dynamoDB object
return dynamoDB;
}

// Public method to add a new attribute to all items in the Super_Volcanoes
// table in DynamoDB and passing the tableName and dynamoDBObject
public void addAttributeToItem(String tableName, DynamoDB dynamoDBObject) {

    // Instantiating the table object by passing the tableName
    Table table = dynamoDBObject.getTable(tableName);

    try {
        // Creating an ArrayList of Map storing the items in the Super_Volcanoes
        // table and iterating through them
        List<Map<String, Object>> itemList = new ArrayList<>();

        // Looping through the items in Super_Volcanoes table
        for (Item item : table.scan()) {

            // Creating a HashMap to store the last eruption period of
            // each super volcano which will then be added to the Super_Volcanoes
            // table in DynamoDB
            Map<String, Object> map = new HashMap<>();

            // Using if-else ladder to condition all super volcanoes in the table
            map.put("Name", item.asMap().get("Name"));
            if (item.asMap().get("Name").toString().toLowerCase().contains(
                "Long Valley Caldera".toLowerCase())) {
                map.put("last_eruption_period", "760,000 years ago");
            } else if (item.asMap().get("Name").toString().toLowerCase().contains(
                "Valles Caldera".toLowerCase())) {
                map.put("last_eruption_period", "1.2 million and 1.6 million years ago");
            } else if (item.asMap().get("Name").toString().toLowerCase().contains(
                "Lake Toba".toLowerCase())) {
                map.put("last_eruption_period", "75,000 years ago");
            } else if (item.asMap().get("Name").toString().toLowerCase().contains(
                "Aira Caldera".toLowerCase())) {
                map.put("last_eruption_period", "22,000 years ago");
            } else if (item.asMap().get("Name").toString().toLowerCase().contains(
                "Taupo Caldera".toLowerCase())) {
                map.put("last_eruption_period", "26,500 years ago");
            } else if (item.asMap().get("Name").toString().toLowerCase().contains(
                "Yellowstone Caldera".toLowerCase())) {
                map.put("last_eruption_period", "640,000 years ago");
            } else {
                map.put("last_eruption_period", "");
            }
        }
    }
}

```



```

        // Adding the map to the itemList
        itemList.add(map);
    }

    // Looping through the itemList
    for (Map<String, Object> map : itemList) {

        // Conditioning the last eruption period attribute in the
        // Super_Volcanoes table in DynamoDB
        if (map.get("last_eruption_period").toString().length() > 1) {

            // Using the UpdateItemSpec object to update the
            // last_eruption_period of the super volcanoes present in the
            // DynamoDB table
            UpdateItemSpec updateItemSpec =
                new UpdateItemSpec().withPrimaryKey("Name",
                    map.get("Name").toString())
                    .withUpdateExpression("set #keyAttribute = :valueAttribute")
                    .withNameMap(new NameMap()
                        .with("#keyAttribute", "last_eruption_period"))
                    .withValueMap(new ValueMap()
                        .withString(":valueAttribute", map.get("last_eruption_period")
                            .toString()).withReturnValues(ReturnValue.ALL_NEW);

            // Updating the item in the Super_Volcanoes table
            table.updateItem(updateItemSpec);
            System.out.println("Items in table updated.");
        }
    }
} catch (Exception e) {
    System.err.println(e.getMessage());
}
}

// Public method to retrieve the items from the Super_Volcanoes
// table in DynamoDB and passing the tableName and DynamoDB object
public void retrieveItem(String tableName, DynamoDB dynamoDBObject) {

    // Instantiating the table object by passing the tableName
    Table table = dynamoDBObject.getTable(tableName);

    try {
        // Create an instance of the TableKeysAndAttributes class that describes
        // a list of primary key values to retrieve from a table.
        TableKeysAndAttributes svTableKeysAndAttributes =
            new TableKeysAndAttributes(tableName);
        svTableKeysAndAttributes.addHashOnlyPrimaryKeys("Name",
            "Long Valley Caldera",
            "Valles Caldera",
            "Lake Toba",
            "Aira Caldera",
            "Taupo Caldera",
            "Yellowstone Caldera");

        // Calling the batchGetItem method by providing the
        // TableKeysAndAttributes objects that you created in the preceding step.
        BatchGetItemOutcome outcome = dynamoDBObject
            .batchGetItem(svTableKeysAndAttributes);

        // Looping through the outcome object
        for (String name : outcome.getTableItems().keySet()) {
            System.out.println("Items in table " + name);
        }
    }
}

```

```
List<Item> items = outcome.getTableItems().get(name);  
for (Item item : items) {  
    System.out.println(item.toJSONPretty());  
}  
}  
} catch (Exception e) {  
    System.err.println(e.getMessage());  
}  
}  
}
```

References

- [1] N. G. Society, "Supervolcanoes and Notable Volcanic," i-cubed, 2013. [Online]. Available: <https://www.arcgis.com/apps/MapJournal/index.html?appid=a546b46a7fb942008455e072c69ea767>.
- [2] A. Dubner, "The World's 6 Known Supervolcanoes," Ranker, 13 October 2018. [Online]. Available: https://www.ranker.com/list/the-world_s-6-known-supervolcanoes/analise.dubner.