Day 29

Employee ID: 201933938

Login ID: iamasif

Name: Shaik Asif

| Dynamo DB | Overview of AWS DynamoDB, Terminology Comparison with SQL, DynamoDB Tables and Naming,Conventions, Data, DynamoDB Consistency Model, DynamoDB Capacity Units, DynamoDB On Demand Capacity, Basics of DynamoDB Partitions, Basics of DynamoDB Indexes, Local Secondary Indexes and Global Secondary Indexes |
| --- | --- |
| Working with DynamoDB using AWS Console3 | Interacting with DynamoDB -Table-level Operations with AWS Console, Item-level Operations with AWS Console, Additional Features in DynamoDB Console |
| Working with DynamoDB using AWS CL | Working with the AWS CLI, Table level Operations with AWS CLI, Write Operations - Item level Operations with AWS CLI, Read Operations - Item level Operations with AWS CLI |
| Working with DynamoDB using AWS SDK | Working with DynamoDB using AWS SDK - Introduction, Table-level Operations with AWS SDK, Write Operations - Item Level Operations with AWS SDK, Conditional Writes - Item Level Operations with AWS SDK, Atomic Counters - Item Level Operations with AWS SDK, Read Operations - Item Level Operations with AWS SDK, Paginated Reads - Item Level Operations with AWS SDK |
| DynamoDB Data Modeling & Best Practices | DynamoDB Architecture, DynamoDB Partitions in Depth, DynamoDB Efficient Key Design, Hot Keys or Hot Partitions, DynamoDB Design Patterns, Multi-value Sorts and Filters, DynamoDB Limits - Error Handling in DynamoDB, Ways to Lower DynamoDB Costs |

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/workbench.settingup.html>

Download NoSQL Workbench for DynamoDB - Amazon DynamoDB

Download the official version of NoSQL Workbench for Amazon DynamoDB. With NoSQL Workbench, you can model schemas, visualize data, and query tables before pushing it all to DynamoDB.

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/workbench.settingup.install.html>

Install NoSQL Workbench for DynamoDB - Amazon DynamoDB

Install the official version of NoSQL Workbench for Amazon DynamoDB. With NoSQL Workbench, you can model schemas, visualize data, and query tables before pushing it all to DynamoDB.

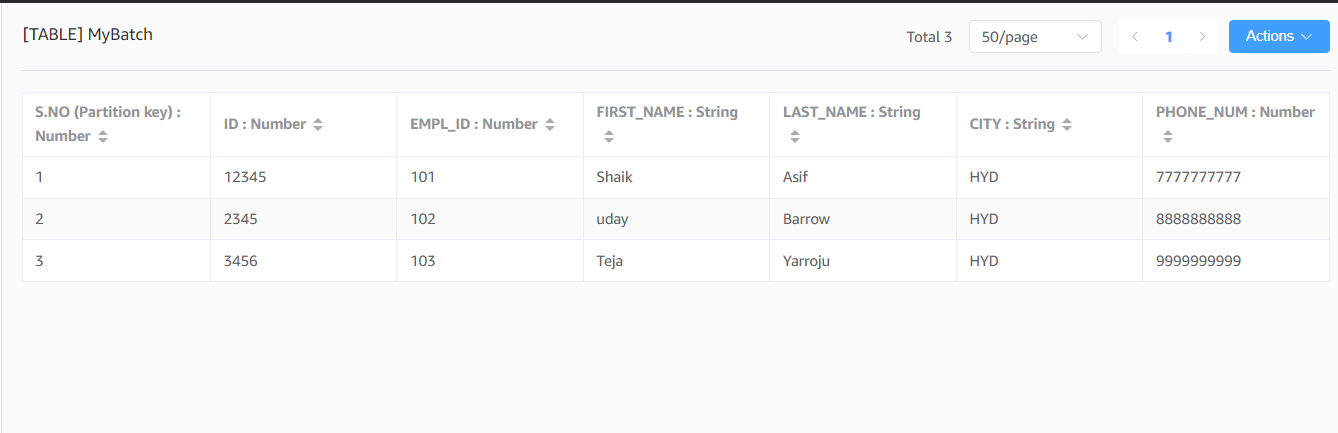
Task 1:

Create a data model -ATLAS Batch1

Add a table - MyBatch

Attributes - id, Empid, f name, l name, city, ph no

12.10 to 12.15



Aws cli download for windows — link pasted…

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

After installing

Open cmd

Type aws - -version

**Installed**

Create table.java

import java.util.Arrays;

import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClient;

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

import com.amazonaws.services.dynamodbv2.document.Table;

import com.amazonaws.services.dynamodbv2.model.AttributeDefinition;

import com.amazonaws.services.dynamodbv2.model.KeySchemaElement;

import com.amazonaws.services.dynamodbv2.model.KeyType;

// throughput

import com.amazonaws.services.dynamodbv2.model.ScalarAttributeType;

public class Createtable {

public static void main(String[] args) throws Exception {

AmazonDynamoDBClient client = new AmazonDynamoDBClient().withEndpoint("http://localhost:8000");

DynamoDB dynamoDB = new DynamoDB(client);

String tableName = "Empoloyee";

try{

Table table = dynamoDB.createTable(tableName, Arrays.asList(new KeySchemaElement("ID", KeyType.HASH), ))

}

}

}

Or same as above.. But below is complete..

import java.util.Arrays;

import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClient;

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

import com.amazonaws.services.dynamodbv2.document.Table;

import com.amazonaws.services.dynamodbv2.model.AttributeDefinition;

import com.amazonaws.services.dynamodbv2.model.KeySchemaElement;

import com.amazonaws.services.dynamodbv2.model.KeyType;

// throughput

import com.amazonaws.services.dynamodbv2.model.ProvisionedThroughput;

import com.amazonaws.services.dynamodbv2.model.ScalarAttributeType;

public class Createtable {

public static void main(String[] args) throws Exception {

AmazonDynamoDBClient client = new AmazonDynamoDBClient().withEndpoint("http://localhost:8000");

DynamoDB dynamoDB = new DynamoDB(client);

String tableName = "Empoloyee";

try{

Table table = dynamoDB.createTable(tableName, Arrays.asList(new KeySchemaElement("ID", KeyType.HASH),

new KeySchemaElement("no", KeyType.Range)

),

Array.asList(new AttributeDefinition("ID", ScalarAttributeType.N),

new AttributeDefinition("no", ScalarAttributeType.S)

),

new ProvisionedThroughput(5L,5L)

);

table.waitForActive();

System.out.println(" table description "+ table.getDescription().getTableStatus());

}catch (Exception ex) {

Susyem.out.println("error "+ ex.getmessage());

}

}

}

Table loading data.java

package com.example;

import java.io.File;

import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClient;

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

import com.amazonaws.services.dynamodbv2.document.Table;

import com.amazonaws.services.dynamodbv2.document.Item;

import com.fasterxml.jackson.JsonFactory; // core.JsonFactory;

import com.fasterxml.jackson.core.JsonParser;

import com.fasterxml.jackson.atabind.JsonNode;

import com.fasterxml.jackson.databind.ObjectMapper;

import com.fasterxml.jackson.databind.node.ObjectNode;

public class EmployeeLoadingData {

public static void main(String[] args) throws Exception {

AmazonDynamoDBClient client = new AmazonDynamoDBClient().withEndpoint("http://localhost:8000");

DynamoDB dynmoDB = new DynamoDB(client);

Table table = dynamoDB.getTable("Employee");

JsonParser parser = new JsonFactory().createParser(new File(Table.json));

JsonNode rootNode = new ObjectMapper().readTree(parser);

Iterator<JsonNode> iterate = rootNode.iterator();

while (iterate.hasNext()) {

currentNode = (ObjectNode) iterate.next();

int ID = currentNode.path("ID").asInt();

String NomenClature = currentNode.path("NomenClature").asText();

try{

table.putItem(new Item().withPrimarykey("ID", ID, "NomenClature", NomenClature));

System.err.println("wow add item in employee table is successful "+ ID + " "+ NomenClature);

}catch(Exception ex) {

System.err.println("sorry cant add item in employee table "+ ID + " "+ NomenClature);

System.err.println(ex.getMessage());

break;

}

}

parser.close();

}

}

Table.json

[

{

"ID" : 1001,

"NomenClature" : "ABC 1001",

"Name" : "Prasunamba"

},

{

"ID" : 1002,

"NomenClature" : "XYZ 1002",

"Name" : "Meher"

},

{

"ID" : 1003,

"NomenClature" : "SSS 1002",

"Name" : "K"

}

]

Info Box:

Updated Excalidraw link till Day 28th

[SUDIPTO (Guest): https://excalidraw.com/#json=xvNUDQ1IwmQtNRFUjagCk,QaLSI7nJet4POU...](https://teams.live.com/l/message/19:meeting_MTY4OTU3ZDgtZmEwZi00MjIzLTk1NzctYTk0MDRmYjI2Mjg5@thread.v2/1755750876802?context=%7B%22contextType%22%3A%22chat%22%7D)

sent on 21 August 2025 10:04

Excalidraw updated at 10.3521st Aug 2025

[https://excalidraw.com/#json=OQ4rEVI5vj4ZpLTGqjgE8,cPKJbB12sGiehttps://excalidraw.com/#json=OQ4rEVI5vj4ZpLTGqjgE8,cPKJbB12sGieNoTUtb95BgNoTUtb95Bg](https://excalidraw.com/#json=OQ4rEVI5vj4ZpLTGqjgE8,cPKJbB12sGieNoTUtb95Bg)