**Creating Databases using MongoDB, DynamoDB, Voldemort Key-Value Distributed Data Store Hbase and Neo4j.**

**Aim:**

To Create Databases using MongoDB, DynamoDB, Voldemort Key-Value Distributed Data Store Hbase and Neo4j.

**Procedure:**

**MongoDB**

**Step 1:** Download MongoDB Community Edition:

* + Go to the [MongoDB Download Center](https://www.mongodb.com/try/download/community) and choose the MSI installer for Windows.

**Step 2:** Install MongoDB:

* + Follow the installation wizard and select **Complete** installation.
  + Ensure the option to install MongoDB as a service is selected.

**Step 3:** Run MongoDB Server:

* + Open Command Prompt and run mongod to start the server. If MongoDB was installed as a service, it should run automatically.

**Step 4:** Use MongoDB Shell:

* + In another Command Prompt window, type mongo to access the MongoDB shell.

**DynamoDB**

**Step 1:** Download DynamoDB Local:

* + Download the DynamoDB Local .zip file from [DynamoDB Local](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DynamoDBLocal.html).

**Step 2:** Extract the ZIP File:

* + Extract it to a directory on your machine, e.g., C:\DynamoDBLocal.

**Step 3:** Run DynamoDB Local:

* + Open Command Prompt, navigate to the DynamoDB directory (cd C:\DynamoDBLocal), and run:

java -Djava.library.path=./DynamoDBLocal\_lib -jar DynamoDBLocal.jar -sharedDb

* + DynamoDB will be accessible at http://localhost:8000.

**Step 4:** Use AWS CLI to Interact (Optional):

* Install the AWS CLI from [AWS CLI Installation](https://aws.amazon.com/cli/).
* Configure it with aws configure, and then you can use commands to create tables and interact with the local DynamoDB.

### Voldemort

**Step 1:** Download Voldemort:

* + Go to [Voldemort GitHub Releases](https://github.com/voldemort/voldemort/releases) and download the latest release.

**Step 2:** Extract Voldemort:

* + Unzip it to a location on your machine, e.g., C:\voldemort.

**Step 3:** Start Voldemort Server:

* + Open Command Prompt, navigate to the Voldemort directory (cd C:\voldemort), and run:

bin\voldemort-server.bat config\single\_node\_cluster

* + This will start the Voldemort server locally.

**Step 4:** Use Voldemort Shell or REST API:

* Use the shell or REST API to interact with the server to insert and retrieve data.

### HBase

**Step 1:** Download HBase:

* + Go to [HBase Downloads](https://hbase.apache.org/downloads.html) and download the latest binary for Windows.

**Step 2:** Install Hadoop (Required):

* + HBase requires Hadoop. Download Hadoop for Windows from [Hadoop Download Page](https://hadoop.apache.org/releases.html).
  + Set up Hadoop by following the steps in [this guide](https://hadoop.apache.org/docs/r3.3.0/hadoop-project-dist/hadoop-common/SingleCluster.html) or install it in pseudo-distributed mode.

**Step 3:** Start HBase:

* + Navigate to the HBase folder and run bin\start-hbase.cmd to start HBase.

**Step 4:** Use HBase Shell:

* + In the HBase folder, open Command Prompt and run hbase shell to access the HBase CLI.

### Neo4j

**Step 1:** Download Neo4j Desktop:

* + Go to the Neo4j Download Center and download Neo4j Desktop for Windows.

**Step 2:** Install Neo4j:

* + Run the downloaded installer and follow the setup instructions.

**Step 3:** Start Neo4j:

* + Open Neo4j Desktop and create a new project.
  + Start the Neo4j database instance from the desktop application.

**Step 4:** Use Neo4j Browser:

* + Access the Neo4j Browser at http://localhost:7474 (default). You can use the browser to run Cypher queries for inserting and querying data.

**Result:**

Thus to Create Databases using MongoDB, DynamoDB, Voldemort Key-Value Distributed Data Store Hbase and Neo4j has be completed successfully

**Writing simple queries to access databases created using MongoDB, DynamoDB, Voldemort Key-Value Distributed Data Store Hbase and Neo4j.**

**Aim:**

To Write simple queries to access databases created using MongoDB, DynamoDB, Voldemort Key-Value Distributed Data Store Hbase and Neo4j.

**Procedure:**

### Prerequisites

1. **Install Node.js**: Make sure you have Node.js installed. You can download it from [Node.js official website](https://nodejs.org/).
2. **Set Up VS Code**: Download and install [Visual Studio Code](https://code.visualstudio.com/).
3. **Install Extensions**:
   * MongoDB: [MongoDB for VS Code](https://marketplace.visualstudio.com/items?itemName=mongodb.mongodb-vscode)
   * Neo4j: [Neo4j VS Code Extension](https://marketplace.visualstudio.com/items?itemName=neo4j.neo4j-vscode)

### Step-by-Step Setup Instructions

#### 1. **Initialize a Node.js Project**

Open VS Code, create a new folder for your project, and initialize a Node.js project.

# Open terminal in VS Code and run:

npm init -y

This will create a package.json file where your project dependencies are listed.

#### 2. **Install Required Packages**

Install the necessary npm packages for each database.

# MongoDB

npm install mongodb

# DynamoDB (AWS SDK)

npm install aws-sdk

# Voldemort (REST calls using axios)

npm install axios

# HBase (hbase-client)

npm install hbase-client

# Neo4j

npm install neo4j-driver

#### 3. **Configure JavaScript Code for Each Database**

For each database, create a separate JavaScript file and add the respective code for operations.

### MongoDB Setup (Using mongodb package)

Create a file named mongodb.js and add the following code:

const { MongoClient } = require('mongodb');

const uri = "mongodb://localhost:27017"; // MongoDB connection string

const client = new MongoClient(uri);

async function run() {

try {

await client.connect();

const database = client.db("testDB");

const students = database.collection("students");

// Insert Data

await students.insertOne({ name: "Alice", age: 22, major: "Computer Science" });

// Retrieve All Documents

const allStudents = await students.find({}).toArray();

console.log("All Students:", allStudents);

// Retrieve Specific Documents

const specificStudent = await students.find({ age: 22 }).toArray();

console.log("Specific Student:", specificStudent);

// Update a Document

await students.updateOne({ name: "Alice" }, { $set: { major: "Mathematics" } });

// Delete a Document

await students.deleteOne({ name: "Alice" });

} finally {

await client.close();

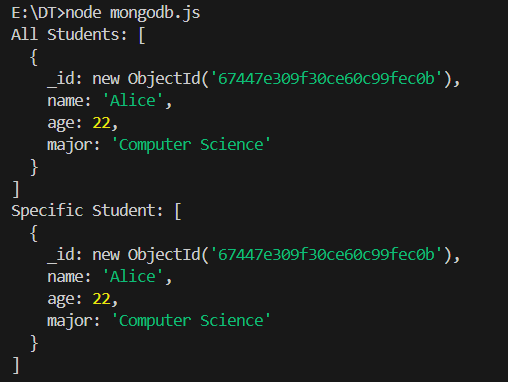
}

}

run().catch(console.dir);

To run this file, open the terminal and execute:  
  
node mongodb.js

**Output:**



### DynamoDB Setup (Using aws-sdk)

Create a file named dynamodb.js and add the following code:

const AWS = require('aws-sdk');

// Configure DynamoDB Local

AWS.config.update({

region: "us-west-2",

endpoint: "http://localhost:8000"

});

const dynamodb = new AWS.DynamoDB.DocumentClient();

const tableName = "Students";

async function run() {

// Insert Data

await dynamodb.put({

TableName: tableName,

Item: { StudentID: "123", Name: "Alice", Age: 22, Major: "Computer Science" }

}).promise();

// Retrieve a Specific Item

const item = await dynamodb.get({

TableName: tableName,

Key: { StudentID: "123" }

}).promise();

console.log("Retrieved Item:", item);

// Update an Item

await dynamodb.update({

TableName: tableName,

Key: { StudentID: "123" },

UpdateExpression: "SET Major = :m",

ExpressionAttributeValues: { ":m": "Mathematics" }

}).promise();

// Delete an Item

await dynamodb.delete({

TableName: tableName,

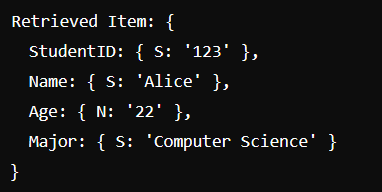
Key: { StudentID: "123" }

}).promise();

}

run().catch(console.error);

Run it using:  
  
node dynamodb.js  
  
**Output:**



### Voldemort Setup (Using axios)

Create a file named voldemort.js and add the following code:

const axios = require('axios');

const baseUrl = "http://localhost:8081/stores/students/";

async function run() {

// Insert Data

await axios.post(baseUrl, {

key: "123",

value: { name: "Alice", age: 22, major: "Computer Science" }

});

// Retrieve Data

const { data } = await axios.get(`${baseUrl}123`);

console.log("Retrieved Data:", data);

// Delete Data

await axios.delete(`${baseUrl}123`);

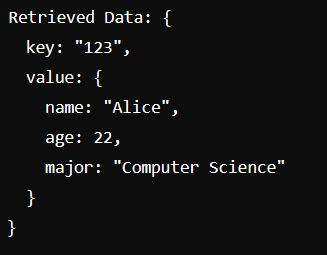
}

run().catch(console.error);

Run it with:

node voldemort.js

**Output:**



### HBase Setup (Using hbase-client)

Create a file named hbase.js and add the following code:

const HBase = require('hbase-client');

const client = HBase.create({

zookeeperHosts: ['localhost:2181'],

zookeeperRoot: '/hbase'

});

async function run() {

// Insert Data

await client.put('students', '1', [

{ column: 'info:name', $: 'Alice' },

{ column: 'info:age', $: '22' },

{ column: 'info:major', $: 'Computer Science' }

]);

// Retrieve Data for a Row

const result = await client.getRow('students', '1');

console.log("Row Data:", result);

// Delete Data

await client.delete('students', '1', 'info:major');

client.close();

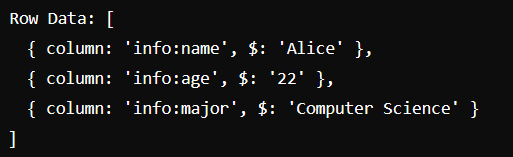
}

run().catch(console.error);

Run with:

node hbase.js

**Output:**



### Neo4j Setup (Using neo4j-driver)

Create a file named neo4j.js and add the following code:

const neo4j = require('neo4j-driver');

const uri = 'bolt://localhost:7687';

const user = 'neo4j';

const password = 'your\_password';

const driver = neo4j.driver(uri, neo4j.auth.basic(user, password));

const session = driver.session();

async function run() {

// Create a Node

await session.run("CREATE (a:Student {name: 'Alice', age: 22, major: 'Computer Science'})");

// Retrieve All Nodes

const result = await session.run("MATCH (s:Student) RETURN s");

result.records.forEach(record => {

console.log("Retrieved Node:", record.get('s').properties);

});

// Update a Node

await session.run("MATCH (s:Student {name: 'Alice'}) SET s.major = 'Mathematics'");

// Delete a Node

await session.run("MATCH (s:Student {name: 'Alice'}) DELETE s");

await session.close();

driver.close();

}

run().catch(console.error);

Run with:

node neo4j.js

**Output:**

****

**Result:**

Thus to Write simple queries to access databases created using MongoDB, DynamoDB, Voldemort Key-Value Distributed Data Store Hbase and Neo4j has been completed successfully