**EXP-NO:4**

**Querying Databases Using Various NoSQL Technologies**

**1. MongoDB**

**Example MongoDB Queries:**

```javascript

// Connect to the MongoDB server

const MongoClient = require('mongodb').MongoClient;

const url = 'mongodb://localhost:27017';

MongoClient.connect(url, { useNewUrlParser: true, useUnifiedTopology: true }, (err, client) => {

if (err) throw err;

// Select the database

const db = client.db('YourDatabaseName');

// Query the collection

const collection = db.collection('YourCollectionName');

// Find all documents

collection.find({}).toArray((err, result) => {

if (err) throw err;

console.log(result);

client.close();

});

});

**Searching with Regular Expressions:**

To perform a search similar to SQL's `LIKE` query, you can use regular expressions in MongoDB:

```javascript

var User = mongoose.model('User');

var searchQuery = {

email: req.query.email,

name: { $regex: req.query.name, $options: 'i' } // Case-insensitive search

};

User.find(searchQuery, function(error, user) {

if (error || user === null) {

return res.status(500).send(error);

}

return res.status(200).send(user);

});

**Alternative Query Using `$where`:**

```javascript

db.myCollection.find({ $where: "this.name.toLowerCase().indexOf('m') >= 0" });

**2. DynamoDB (AWS)**

**Example DynamoDB Queries:**

javascript

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

AWS.config.update({

region: 'your-region',

accessKeyId: 'your-access-key-id',

secretAccessKey: 'your-secret-access-key',

});

const dynamodb = new AWS.DynamoDB();

// Query the DynamoDB table

const params = {

TableName: 'YourTableName',

KeyConditionExpression: 'YourPartitionKey = :value',

ExpressionAttributeValues: {

':value': { S: 'your-value' },

},

};

dynamodb.query(params, (err, data) => {

if (err) console.error(err);

else console.log(data);

});

**3. Voldemort Key-Value Distributed Data Store**

**Example Voldemort Queries:**

Voldemort is a key-value store, so querying involves retrieving data based on keys.

**Assuming you have a client configured**

StoreClient<String, String> client = new SocketStoreClient<>(new ClientConfig().setBootstrapUrls("tcp://localhost:6666"));

**Get value by key**

String value = client.getValue("your\_key");

System.out.println(value);

**4. HBase**

**Example HBase Queries:**

Configuration config = HBaseConfiguration.create();

Connection connection = ConnectionFactory.createConnection(config);

Table table = connection.getTable(TableName.valueOf("YourTableName"));

**Get by row key**

Get get = new Get(Bytes.toBytes("your-row-key"));

Result result = table.get(get);

**Process the result**

for (Cell cell : result.rawCells()) {

System.out.println(Bytes.toString(CellUtil.cloneRow(cell)) +

" " + Bytes.toString(CellUtil.cloneFamily(cell)) +

":" + Bytes.toString(CellUtil.cloneQualifier(cell)) +

" " + Bytes.toString(CellUtil.cloneValue(cell)));

}

table.close();

connection.close();

**5. Neo4j**

**Example Neo4j Queries:**

Neo4j uses the Cypher query language.

`cypher

**Create a node**

CREATE (n:Person {name: 'John Doe', age: 30})

**Retrieve all nodes**

MATCH (n) RETURN n

**Retrieve nodes with a specific property value**

MATCH (n:Person {name: 'John Doe'}) RETURN n