Code:-

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

const url = 'mongodb://127.0.0.1:27017';

const client = new MongoClient(url);

async function createNewDatabase(dbName) {

  try {

    await client.connect().then(() => console.log("Connected With Server")).catch(err => console.log("Unable to connect"));

    var db = client.db(dbName);

    // var collection = db.createCollection('employeeCollection');

    console.log(`Created new database '${dbName}'`);

  } catch (err) {

    console.error(err);

  } finally {

    await client.close();

  }

}

//--------------------Insert data-----------------------------------------

async function insertData(dbName, collectionName) {

  //establish the connection with MongoDB database

  await client.connect().then(() => console.log("Again Connected")).catch(err => console.log("Unable to connect to database"));

  //We take database reference

  const myDB = client.db(dbName);

  /\* we cannot use here createCollection() because it create only one time collection if second time same code

  we run then give error So we us collection().

  -->if collection is not created then automatically collection() can create collection\*/

  const myColl = myDB.collection(collectionName);

  try {

    const data = [

      { name: 'Vasu', age: 20 },

      { name: 'Ashneer', age: 30 },

      { name: 'Piyush', age: 40 },

      { name: 'Tata', age: 85 }

    ];

    const insertManyresult = await myColl.insertMany(data);

    let ids = insertManyresult.insertedIds;

    console.log(`${insertManyresult.insertedCount} documents were inserted.`); //count the number of inserted document

    for (let id of Object.values(ids)) {

      console.log(`Inserted a document with id ${id}`);

    }

  } catch (e) {

    console.log(e);

    // console.log(`A MongoBulkWriteException occurred, but there are successfully processed documents.`);

    let ids = e.result.result.insertedIds;

    for (let id of Object.values(ids)) {

      console.log(`Processed a document with id ${id.\_id}`);

    }

    console.log(`Number of documents inserted: ${e.result.result.nInserted}`);

  } finally {

    await client.close();

  }

}

//----------------------Update data-------------------------------------

async function updateData(dbName, collectionName) {

  //establish the connection with MongoDB database

  try {

    await client.connect().then(() => console.log("Again Again Connected")).catch(err => console.log("Unable to connect to database"));

    const myDB = client.db(dbName);

    const myColl = myDB.collection(collectionName);

    const filter = { name: 'Vasu' };

    // update the value of the 'quantity' field to 5

    const updateDocument = {

      $set: { age: 16 },

    };

    const result = await myColl.updateOne(filter, updateDocument);

  } catch (e) {

    console.log(e);

  } finally {

    await client.close();

  }

}

//---------------Delete data-------------------------------------

async function deleteData(dbName, collectionName) {

  //establish the connection with MongoDB database

  try {

    await client.connect().then(() => console.log("Again Again Again Connected")).catch(err => console.log("Unable to connect to database"));

    const myDB = client.db(dbName);

    const myColl = myDB.collection(collectionName);

    const deleteData = { name: 'Vasu', age: 16 };

    const result = await myColl.deleteOne(deleteData);   //deleteOne() take parameter as Object like JSON, javascript object

  } catch (e) {

    console.log(e);

  } finally {

    await client.close();

  }

}

createNewDatabase('Employee').then(() => console.log("Database Created")).catch(err => console.log("Unable to Create Database"));

insertData('Employee', 'employeeCollection').then(() => console.log("Inserted")).catch(err => console.log("Unable to Insert"));

updateData('Employee', 'employeeCollection').then(() => console.log("Updated")).catch(err => console.log("Unable to Update"));

deleteData('Employee', 'employeeCollection').then(() => console.log("Deleted")).catch(err => console.log("Unable to Update"));

Output:-







