Operar sobre diferentes motores

CREATE TABLE [dbo].[Clientes](

[id] [int] NOT NULL IDENTITY(1,1),

[Nombre] [varchar](50) NOT NULL,

[Apellido] [varchar](50) NOT NULL,

[Edad] [smallint] NOT NULL

CONSTRAINT [PK\_Clientes] PRIMARY KEY CLUSTERED

([id] ASC)) ON [PRIMARY]

CREATE TABLE [dbo].[ClientesTelefonos](

[id] [int] NOT NULL IDENTITY(1,1),

[idCliente] [int] NOT NULL,

[telefono] [varchar](50) NOT NULL,

CONSTRAINT [PK\_ClientesTelefonos] PRIMARY KEY CLUSTERED

([id] ASC)) ON [PRIMARY]

ALTER TABLE [dbo].[ClientesTelefonos] WITH CHECK ADD CONSTRAINT [FK\_ClientesTelefonos\_Clientes] FOREIGN KEY([idCliente]) REFERENCES [dbo].[Clientes] ([id])

INSERT INTO [dbo].[Clientes]([Nombre],[Apellido],[Edad]) VALUES('Cliente 1','Apellido 1',34)

INSERT INTO [dbo].[Clientes]([Nombre],[Apellido],[Edad]) VALUES('Cliente 2','Apellido 2',56)

INSERT INTO [dbo].[Clientes]([Nombre],[Apellido],[Edad]) VALUES('Cliente 3','Apellido 3',28)

INSERT INTO [dbo].[ClientesTelefonos]([idCliente],[telefono]) VALUES(1, '+541122223333')

INSERT INTO [dbo].[ClientesTelefonos]([idCliente],[telefono]) VALUES(1, '+541145645645')

INSERT INTO [dbo].[ClientesTelefonos]([idCliente],[telefono]) VALUES(2, '+541132132132')

INSERT INTO [dbo].[ClientesTelefonos]([idCliente],[telefono]) VALUES(3, '+541111223344')

INSERT INTO [dbo].[ClientesTelefonos]([idCliente],[telefono]) VALUES(3, '+541155443322')

const sql = require('mssql')

const config = {

    server: 'LUCIANOP',

    user: 'redusers',

    password: 'redusers',

    database: 'MSSQLNodeJS',

    options: {

        instanceName: 'SQLSERVER2019',

        trustedConnection: true,

        enableArithAbort: true

    }

};

const pool = new sql.ConnectionPool(config);

const poolConnect = pool.connect();

async function Consulta() {

    await poolConnect;

    try {

        const consulta = `SELECT cli.Nombre + ', ' + cli.Apellido AS Cliente,

                                 cli.Edad, clitel.telefono as Teléfono

                          FROM Clientes cli

                          INNER JOIN ClientesTelefonos clitel ON cli.id = clitel.idCliente`;

        let resultado = await pool.request().query(consulta);

        console.table(resultado.recordset)

    } catch (err) {

        console.log(err);

    }

}

Consulta();

async function InsertarCliente() {

    await poolConnect;

    const transaction = new sql.Transaction(pool);

    transaction.begin(err => {

        if (err) {

            console.log('Ocurrió un error al iniciar la transacción', err);

            return;

        }

        const cliente = {

            Nombre: '\'Cliente 4\'',

            Apellido: '\'Apellido 4\'',

            Edad: '44'

        };

        const request = new sql.Request(transaction)

        const consulta = `INSERT INTO Clientes(Nombre, Apellido, Edad)

                          VALUES(${cliente.Nombre},${cliente.Apellido},${cliente.Edad})`;

        console.log(consulta);

        request.query(consulta,

            (err, result) => {

                if (err) {

                    console.log('Ocurrió un error al iniciar ejecutar el insert', err);

                    return;

                }

                transaction.commit(err => {

                    if (err) {

                        console.log('Ocurrió un error al finalizar la transacción', err);

                    } else {

                        console.log('La transacción finalizó satisfactoriamente');

                    }

                });

            });

    });

}

InsertarCliente();

CREATE ROLE redusers WITH

LOGIN

NOSUPERUSER

NOCREATEDB

NOCREATEROLE

INHERIT

NOREPLICATION

CONNECTION LIMIT -1

PASSWORD 'redusers';

GRANT SELECT, USAGE ON SEQUENCE clientes\_seq TO redusers;

GRANT SELECT, USAGE ON SEQUENCE clientestelefonos\_seq TO redusers;

CREATE SEQUENCE Clientes\_seq;

CREATE TABLE Clientes(

id int NOT NULL DEFAULT NEXTVAL ('Clientes\_seq'),

Nombre varchar(50) NOT NULL,

Apellido varchar(50) NOT NULL,

Edad smallint NOT NULL,

CONSTRAINT PK\_Clientes PRIMARY KEY (id)

);

CREATE SEQUENCE ClientesTelefonos\_seq;

CREATE TABLE ClientesTelefonos(

id int NOT NULL DEFAULT NEXTVAL ('ClientesTelefonos\_seq'),

idCliente int NOT NULL,

telefono varchar(50) NOT NULL,

CONSTRAINT PK\_ClientesTelefonos PRIMARY KEY (id)

);

ALTER TABLE ClientesTelefonos ADD CONSTRAINT "FK\_ClientesTelefonos\_Clientes" FOREIGN KEY (idCliente) REFERENCES Clientes (id) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT;

INSERT INTO clientes(nombre, apellido, edad) VALUES('Cliente 1','Apellido 1',34);

INSERT INTO clientes(nombre, apellido, edad) VALUES('Cliente 2','Apellido 2',56);

INSERT INTO clientes(nombre, apellido, edad) VALUES('Cliente 3','Apellido 3',28);

INSERT INTO clientesTelefonos(idCliente,telefono) VALUES(1, '+541122223333');

INSERT INTO clientesTelefonos(idCliente,telefono) VALUES(1, '+541145645645');

INSERT INTO clientesTelefonos(idCliente,telefono) VALUES(2, '+541132132132');

INSERT INTO clientesTelefonos(idCliente,telefono) VALUES(3, '+541111223344');

INSERT INTO clientesTelefonos(idCliente,telefono) VALUES(3, '+541155443322');

const { Pool } = require('pg')

const config = {

    user: 'redusers',

    host: 'localhost',

    database: 'PostgreSQLNodeJS',

    password: 'redusers',

    port: 5432,

};

const pool = new Pool(config);

async function Consulta() {

    try {

        const consulta = `SELECT cli.Nombre || ', ' || cli.Apellido AS Cliente,

                                 cli.Edad, clitel.telefono as Teléfono

                          FROM Clientes cli

                          INNER JOIN ClientesTelefonos clitel ON cli.id = clitel.idCliente`;

        let resultado = await pool.query(consulta);

        console.table(resultado.rows);

    } catch (err) {

        console.log(err);

    }

}

Consulta();

async function InsertarCliente() {

    const client = await pool.connect()

    const cliente = {

        Nombre: 'Cliente 4',

        Apellido: 'Apellido 4',

        Edad: '44'

    };

    try {

        await client.query('BEGIN');

        const consulta = `INSERT INTO Clientes(Nombre, Apellido, Edad) VALUES($1,$2,$3) RETURNING id`;

        const res = await client.query(consulta, [cliente.Nombre, cliente.Apellido, cliente.Edad]);

        await client.query('COMMIT');

        console.log('La transacción finalizó satisfactoriamente');

    } catch (e) {

        await client.query('ROLLBACK');

        console.log('Ocurrió un error al finalizar la transacción', e);

    } finally {

        client.release();

    }

}

InsertarCliente();

CREATE TABLE `redusers\_schema`.`clientes` (

`id` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(50) NOT NULL,

`apellido` VARCHAR(50) NOT NULL,

`edad` TINYINT NOT NULL,

PRIMARY KEY (`id`));

CREATE TABLE `redusers\_schema`.`clientestelefonos` (

`id` INT NOT NULL AUTO\_INCREMENT,

`idCliente` INT NOT NULL,

`telefono` VARCHAR(50) NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_clientes\_id\_idx` (`idCliente` ASC) VISIBLE,

CONSTRAINT `fk\_clientes\_id`

FOREIGN KEY (`idCliente`)

REFERENCES `redusers\_schema`.`clientes` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

INSERT INTO clientes(nombre, apellido, edad) VALUES('Cliente 1','Apellido 1',34);

INSERT INTO clientes(nombre, apellido, edad) VALUES('Cliente 2','Apellido 2',56);

INSERT INTO clientes(nombre, apellido, edad) VALUES('Cliente 3','Apellido 3',28);

INSERT INTO clientesTelefonos(idCliente,telefono) VALUES(1, '+541122223333');

INSERT INTO clientesTelefonos(idCliente,telefono) VALUES(1, '+541145645645');

INSERT INTO clientesTelefonos(idCliente,telefono) VALUES(2, '+541132132132');

INSERT INTO clientesTelefonos(idCliente,telefono) VALUES(3, '+541111223344');

INSERT INTO clientesTelefonos(idCliente,telefono) VALUES(3, '+541155443322');

var mysql = require('mysql');

const config = {

    connectionLimit: 50,

    host: 'localhost',

    user: 'redusers',

    password: 'redusers',

    database: `redusers\_schema`

};

const pool = mysql.createPool(config);

function Consulta() {

    const consulta = `SELECT CONCAT(cli.nombre, ", ", cli.apellido) AS Cliente,

                             cli.edad, clitel.telefono as Teléfono

                      FROM Clientes cli

                      INNER JOIN ClientesTelefonos clitel ON cli.id = clitel.idCliente;`;

    pool.query(consulta, function (error, results, fields) {

        if (error) throw error;

        console.table(results);

    });

}

Consulta();

function InsertarCliente() {

    pool.getConnection(function (err, connection) {

        connection.beginTransaction(function (err) {

            if (err) { throw err; }

            const cliente = {

                nombre: 'Cliente 4',

                apellido: 'Apellido 4',

                edad: 44

            };

            connection.query('INSERT INTO clientes SET ?', cliente, function (error, results, fields) {

                if (error) {

                    return connection.rollback(function () {

                        throw error;

                    });

                }

                connection.commit(function (err) {

                    if (err) {

                        return connection.rollback(function () {

                            throw err;

                        });

                    }

                    console.log('success!');

                });

            });

        });

    });

}

InsertarCliente();

[

{

"nombre":"Cliente 1",

"apellido":"Apellido 1",

"edad":"34",

"telefonos":[

{"telefono":"+541122223333"},

{"telefono":"+541145645645"}

]

},

{

"nombre":"Cliente 2",

"apellido":"Apellido 2",

"edad":"56",

"telefonos":[

{"telefono":"+541132132132'"}

]

},

{

"nombre":"Cliente 3",

"apellido":"Apellido 3",

"edad":"28",

"telefonos":[

{"telefono":"+541111223344'"},

{"telefono":"+541155443322"}

]

}

]

var mongoose = require('mongoose');

var Schema = mongoose.Schema;

var telefonoSchema = new Schema({

    telefono: String

});

var clienteSchema = new Schema({

    nombre: String,

    apellido: String,

    edad: String,

    telefonos: [telefonoSchema]

});

mongoose.connect('mongodb://localhost:27017/MongoDBNodeJS', {

        useNewUrlParser: true,

        useUnifiedTopology: true

    },

    (err, res) => {

        if (err)

            throw (err);

        else {

            console.log('La conexión a la base de datos se estableció satisfactoriamente.');

        }

    });

const cliModel = mongoose.model('Clientes', clienteSchema);

const telModel = mongoose.model('Telefonos', telefonoSchema);

function Consulta() {

    var query = cliModel.find().lean();

    query.exec(function(err, docs) {

        const docsJSON = docs.map((doc) => {

            doc.telefonos = doc.telefonos.map((tel) => {

                return tel.telefono;

            })

            return doc;

        });

        console.table(docsJSON);

    });

}

Consulta();

function AgregarCliente() {

    let doc = new cliModel();

    let tel1 = new telModel(),

        tel2 = new telModel()

    tel1.telefono = '+541122223333';

    tel2.telefono = '+548989898989';

    doc.telefonos.push(tel1);

    doc.telefonos.push(tel2);

    doc.nombre = 'Cliente 4';

    doc.apellido = 'Apellido 4';

    doc.edad = 44;

    doc.save(null, (err, doc) => {

        if (err) {

            console.log('Ocurrió un error al agregar el cliente', err);

            return;

        }

        console.log('El cliente fue agregado satisfactoriamente', doc);

    });

}

AgregarCliente();