Fernando Saez

Usando mysql desde Node.js

1. Crear una Base de Datos en Mysql
   - Instalar Mysql
   - Instalar alguna herramienta de gestión (Mysql workbench, Phpmyadmin)
2. npm install mysql (o mysql2)
3. Establecer una conexión a la BD

Conectandonos a mysql

```
var mysql = require('mysql');
var connection = mysql.createConnection({
   host: 'localhost',
   user: 'root',
   password: '',
   database: 'node_mysql',
   port: 3306
});
connection.connect(function(error){
   if(error){
      throw error;
   }else{
      console.log('Conexion correcta.');
   }
});
connection.end();
```

Pool de conexiones

```
var mysql = require('mysql’);

var poolconnection = mysql.createPool({
 connectionLimit : 10,
 host : 'localhost',
 user : 'root',
 password : '',
 database : 'mydb',
});

poolconnection.query(query, params, function (error, results,
fields) {
    //Hacer cosas con la BD
});
```

Insert – Update y Delete

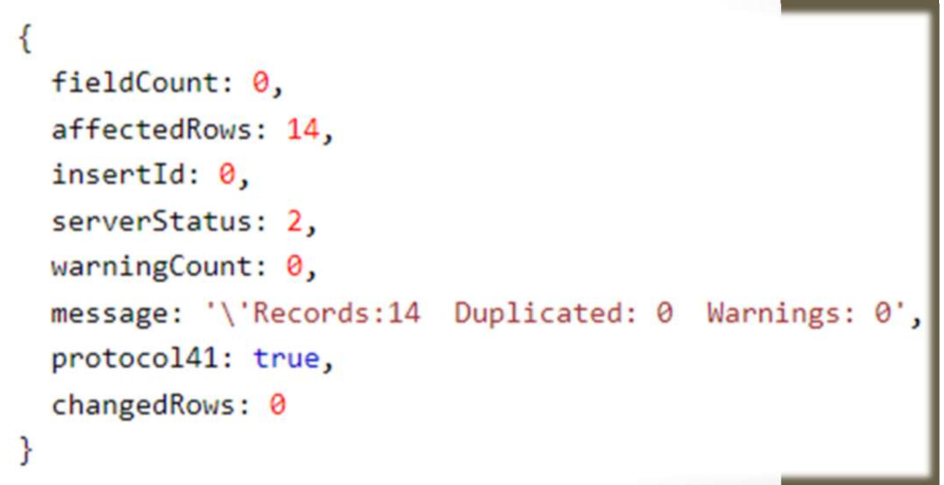
```
//Establecemos la conección en la variable con

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "INSERT INTO cliente (nombre, domicilio) VALUES (‘Tienda Carlitos’, ‘Rivadavia 984')";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log(result.affectedRows + ‟ cliente insertado");
  });
});

//finalizar la conexión
```

Sentencias Preparadas

```
//Establecemos la conexión en la variable connection

var query = connection.query('INSERT INTO personaje(nombre,
apellido, biografia) VALUES(?, ?, ?)', ['Homero', 'Simpson',
'Esposo de Marge'], function(error, result){
   if(error){
      throw error;
   }else{
      console.log(“Personajes insertados: “ +result.affectedRows);
   }
 }
);

//finalizar la conexión
```

Insert - múltiples filas

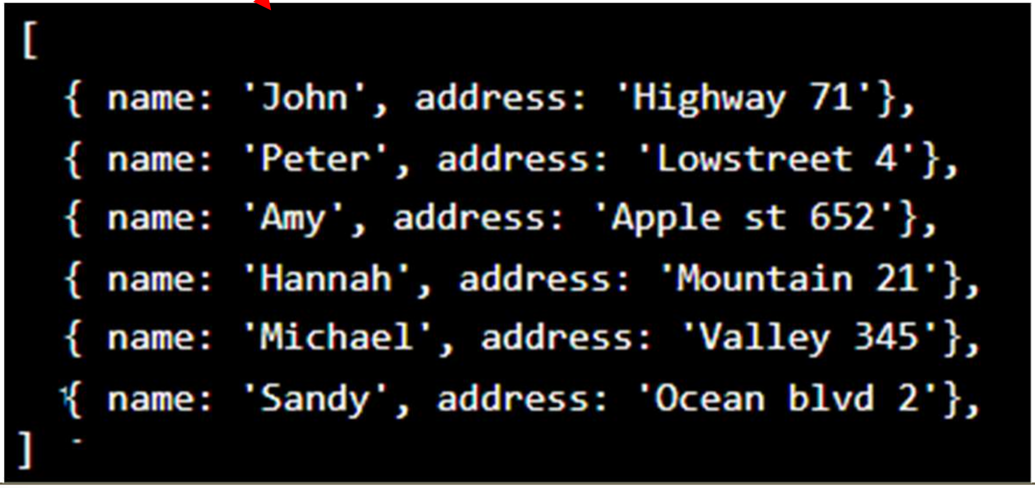
```
var con = mysql.createConnection({multipleStatements: true});

con.connect(function(err) {
  if (err) throw err;
  console.log("Conectedo!");
  var sql = "INSERT INTO customers (name, address) VALUES ?";
  var values = [
    ['John', 'Highway 71'],
    ['Peter', 'Lowstreet 4'],
    ['Amy', 'Apple st 652'],
    ['Hannah', 'Mountain 21'],
    ['Michael', 'Valley 345'],
    ['Sandy', 'Ocean blvd 2'],
    ['Betty', 'Green Grass 1']
];
  con.query(sql, [values], function (err, result) {
    if (err) throw err;
    console.log(“Número de registros insertados: " + result.affectedRows);
  });
});
//finalizar la conexión
```



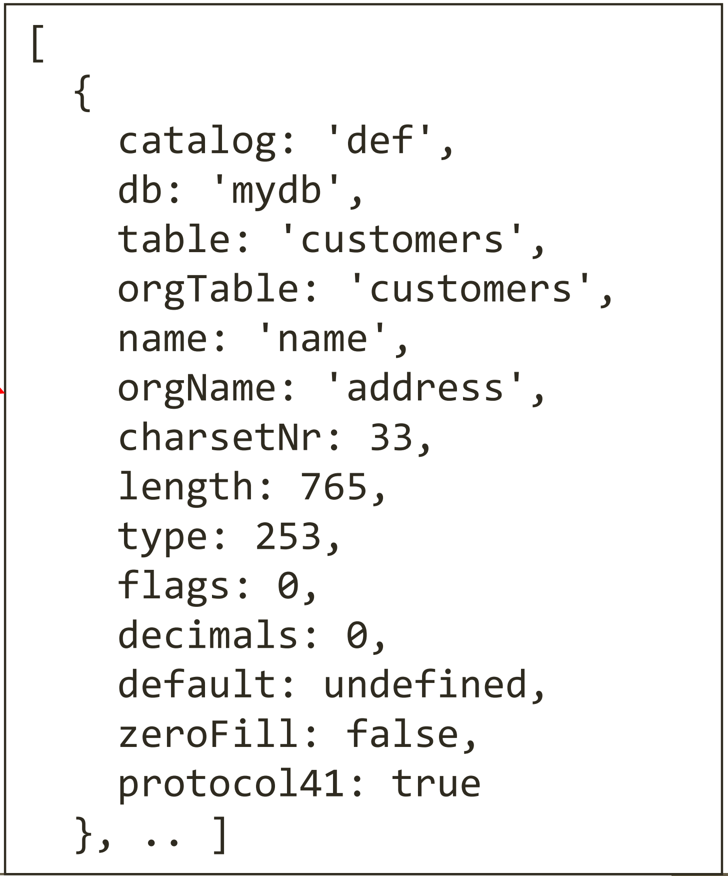
```
{
  fieldCount: 0,
  affectedRows: 14,
  insertId: 0,
  serverStatus: 2,
  warningCount: 0,
  message: '\'Records:14  Duplicated: 0  Warnings: 0',
  protocol41: true,
  changedRows: 0
}
```

Recuperando datos de los campos

```
//Establecemos la conexión en la variable connection
con.connect(function(err) {
  if (err) throw err;
  con.query("SELECT name, address FROM customers", function (err,
result, fields) {
    if (err) throw err;
    console.log(fields);
  });
});//finalizar la conexión
```



```
[
  { name: 'John', address: 'Highway 71'},
  { name: 'Peter', address: 'Lowstreet 4'},
  { name: 'Amy', address: 'Apple st 652'},
  { name: 'Hannah', address: 'Mountain 21'},
  { name: 'Michael', address: 'Valley 345'},
  { name: 'Sandy', address: 'Ocean blvd 2'},
]
```



```
[
  {
    catalog: 'def',
    db: 'mydb',
    table: 'customers',
    orgTable: 'customers',
    name: 'name',
    orgName: 'address',
    charsetNr: 33,
    length: 765,
    type: 253,
    flags: 0,
    decimals: 0,
    default: undefined,
    zeroFill: false,
    protocol41: true
  }, .. ]
```

Select

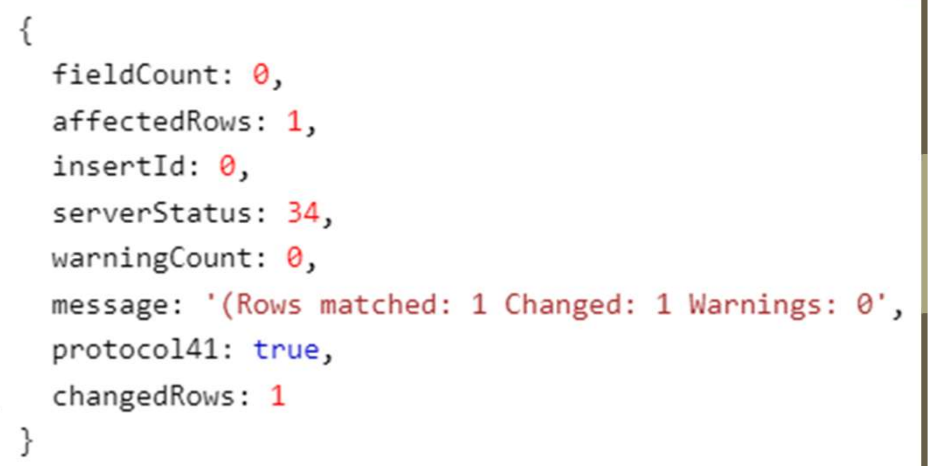
```
//Establecemos la conexión en la variable connection
var query = connection.query('SELECT nombre, apellido, biografia FROM
personaje WHERE personaje_id = ?', [1], function(error, resultado){
      if(error){
         throw error;
      }else{
           if(resultado.length > 0){
           console.log(resultado[0].nombre+’ '+ resultado[0].apellido);
         }else{
            console.log('Registro no encontrado');
         }
      }
   }
);
//finalizar la conexión
```

Delete

```
//Establecemos la conección en la variable connection

con.connect(function(err) {
  if (err) throw err;
  var sql = "DELETE FROM customers WHERE address = 'Mountain 21'";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Number of records deleted: " + result.affectedRows);
  });
});

//finalizar la conexión
```

Update

```
//Establecemos la conección en la variable connection

con.connect(function(err) {
  if (err) throw err;
  var sql = "UPDATE customers SET address = 'Canyon 123' WHERE address = 'Valley 345'";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log(result.affectedRows + " record(s) updated");
  });
});

//finalizar la conexión
```



```
{
  fieldCount: 0,
  affectedRows: 1,
  insertId: 0,
  serverStatus: 34,
  warningCount: 0,
  message: '(Rows matched: 1 Changed: 1 Warnings: 0',
  protocol41: true,
  changedRows: 1
}
```

Ejecutando Stored Procedure

```
let mysql = require('mysql');
let config = require('./config.js');

let connection =
mysql.createConnection(config);

let sql = `CALL filterTodo(?)`;

connection.query(sql, true, (error, results,
fields) => {
  if (error) {
    return console.error(error.message);
  }
  console.log(results[0]);
});
//finalizar la conexión
```

```
DELIMITER $$
CREATE PROCEDURE
`filterTodo`(IN done
BOOLEAN)
BEGIN
    SELECT * FROM
todos WHERE
completed = done;
END$$
DELIMITER ;
```

```
//config.js
let config = { host
: 'localhost', user
: 'root', password:
'', database:
'todoapp' };
module.exports =
config;
```

Transacciones

```
connection.beginTransaction(function (err) {
  if (err) { throw err; }
  connection.query('INSERT INTO posts SET title=?', title, function (error, results, fields) {
    if (error) {
      return connection.rollback(function() {
        throw error;
      });
    }
 var log = 'Post ' + results.insertId + ' added';
```

```
   connection.query('INSERT INTO log SET data=?', log, 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!');});});
 });});
```

Mysql y async-await

```
const db = require("../connection");
const util = require("util");

// promise wrapper para soportar promesas (async await) con MYSQL
db.query = util.promisify(db.query).bind(db);

module.exports = {
 getUser: async (req, res) => {
    let queryString = `SELECT * from users WHERE id = 89`;
    const [user]= await db.query(queryString).catch(err=>{throw err});
    res.json(user);
 }
}
```