

## Módulo 4 - Día 1

### Reto 2

Ejemplo de INSERT en tabla teacher desde Workbench

[Review the SQL Script to be Applied on the Database](#)

```
1  INSERT INTO `codenotch`.`teacher` (`first_name`, `last_name`) VALUES ('Juan', 'Rodríguez');
2  INSERT INTO `codenotch`.`teacher` (`first_name`, `last_name`) VALUES ('Pedro', 'García');
```

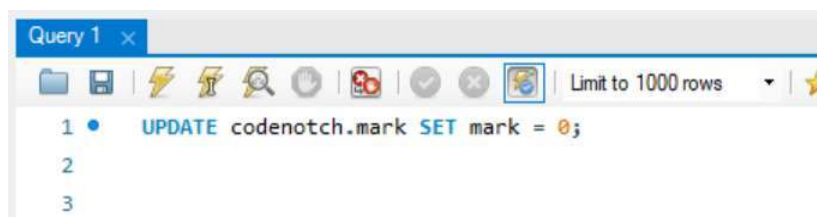
Ejemplo de INSERT en tabla subject desde nodejs

```
14
15 // json provisional para hacer los 10 INSERT de subject
16 let subjects = [
17   { title: 'CSS' },
18   { title: 'jQuery' },
19   { title: 'nodejs' },
20   { title: 'MySQL' },
21   { title: 'javascript' },
22   { title: 'Programación' },
23   { title: 'html' },
24   { title: 'Typescript' },
25   { title: 'Angular' },
26   { title: 'MongoDB' }
27 ]
28
29 // Peticiones query a la bbdd
30 subjects.forEach((subject) => {
31   let sql = "INSERT INTO `codenotch`.`subject` (`title`) VALUES ('" + subject.title + "')";
32   codenotchDB.query(sql, (error, result) => {
33     if (!error) {
34       console.log('INSERT correcto');
35       console.log(result);
36     } else {
37       console.log(error)
38     }
39   })
40 })
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
affectedRows: 1,
insertId: 10,
info: '',
serverStatus: 2,
warningStatus: 0
}
```

Setear todas las notas a 0 desde WorkBench



Setear todas las notas a 0 desde nodejs

```
41 // Setear todas las notas de los alumnos a 0
42 let sql = "UPDATE `mark` SET `mark` = '0'";
43 codenotchDB.query(sql, (error, result) => {
44   if (!error) {
45     console.log('UPDATE correcto');
46     console.log(result);
47   } else {
48     console.log(error)
49   }
50 })
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
affectedRows: 10,
insertId: 0,
info: 'Rows matched: 10 Changed: 10 Warnings: 0',
serverStatus: 34,
warningStatus: 0,
changedRows: 10
}
```

Obtener el nombre y apellido de todos los estudiantes desde WorkBench

Query 1 x

Limit to 1000 rows

```
1 SELECT first_name, last_name FROM codenotch.student;
2
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	first_name	last_name
▶	Ander	Jurado
	Juan José	Cabrera
	Jose	Silva
	Pedro	Abenza
	Pascual	Vera
	Edgar	Rodríguez
	Laura	Henríquez
	Rafael	Bravo
	Rubén	Carballo
	Javier	Fernández

Obtener el nombre y apellido de todos los estudiantes desde nodejs

```
52 // Obtener nombre y apellido de todos los estudiantes
53 let sql = "SELECT `first_name`, `last_name` FROM `codenotch`.`student`";
54 codenotchDB.query(sql, (error, result) => {
55   if (!error) {
56     console.log('SELECT correcto');
57     console.log(result);
58   } else {
59     console.log(error)
60   }
61 })
62
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

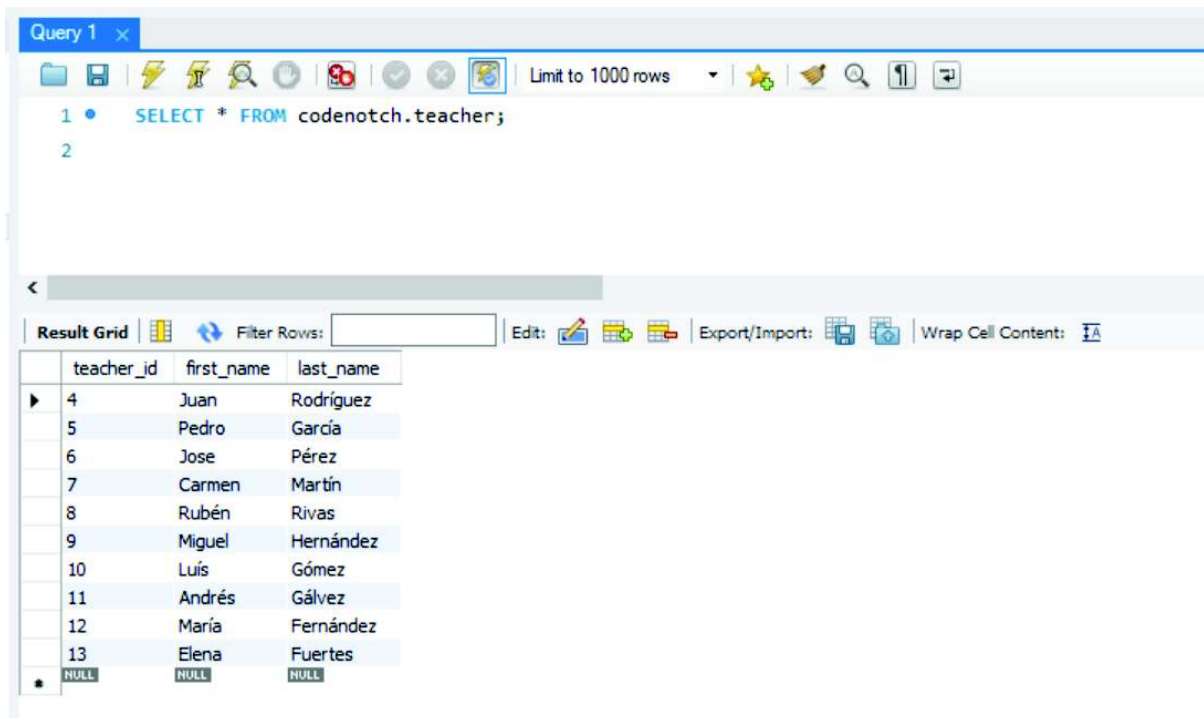
PS C:\Users\ajura\Documents\codenotch\Bootcamp\Modulo04\mysql\dia1\reto2> node .\index.js

Conectado a la bbdd codenotch

SELECT correcto

```
[
  { first_name: 'Ander', last_name: 'Jurado' },
  { first_name: 'Juan José', last_name: 'Cabrera' },
  { first_name: 'Jose', last_name: 'Silva' },
  { first_name: 'Pedro', last_name: 'Abenza' },
  { first_name: 'Pascual', last_name: 'Vera' },
  { first_name: 'Edgar', last_name: 'Rodríguez' },
  { first_name: 'Laura', last_name: 'Henríquez' },
  { first_name: 'Rafael', last_name: 'Bravo' },
  { first_name: 'Rubén', last_name: 'Carballo' },
  { first_name: 'Javier', last_name: 'Fernández' }
]
```

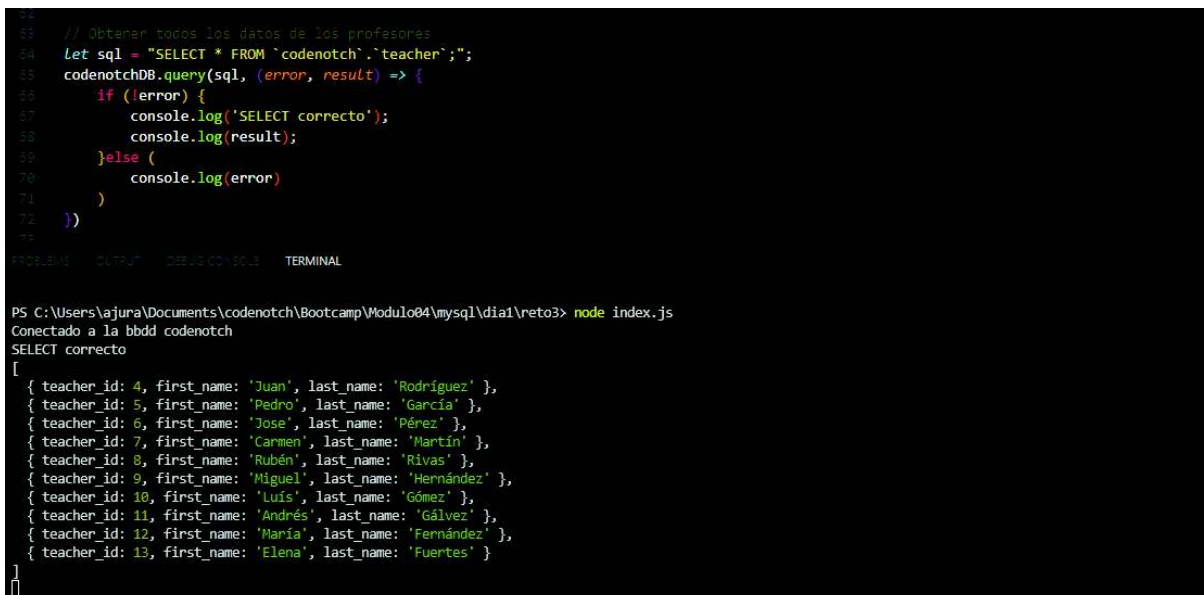
Obtener todos los datos de los profesores desde Workbench



The screenshot shows the MySQL Workbench interface. At the top, a query editor window titled 'Query 1' contains the SQL statement: `SELECT * FROM codenotch.teacher;`. Below the query editor, the 'Result Grid' tab is active, displaying the results of the query in a table format. The table has three columns: `teacher_id`, `first_name`, and `last_name`. The results are as follows:

teacher_id	first_name	last_name
4	Juan	Rodríguez
5	Pedro	García
6	Jose	Pérez
7	Carmen	Martín
8	Rubén	Rivas
9	Miguel	Hernández
10	Luis	Gómez
11	Andrés	Gálvez
12	María	Fernández
13	Elena	Fuertes
NULL	NULL	NULL

Obtener todos los datos de los profesores desde nodejs



The screenshot shows a Node.js application running a MySQL query. The code is as follows:

```
// Obtener todos los datos de los profesores
let sql = "SELECT * FROM `codenotch`.`teacher`";
codenotchDB.query(sql, (error, result) => {
  if (!error) {
    console.log('SELECT correcto');
    console.log(result);
  } else {
    console.log(error)
  }
})
```

The application is running in a terminal window. The output shows the command `node index.js` being executed, and the results of the query are displayed in a JSON array format:

```
PS C:\Users\ajuna\Documents\codenotch\Bootcamp\Modulo04\mysql\dia1\reto3> node index.js
Conectado a la bbdd codenotch
SELECT correcto
[
  { teacher_id: 4, first_name: 'Juan', last_name: 'Rodríguez' },
  { teacher_id: 5, first_name: 'Pedro', last_name: 'García' },
  { teacher_id: 6, first_name: 'Jose', last_name: 'Pérez' },
  { teacher_id: 7, first_name: 'Carmen', last_name: 'Martín' },
  { teacher_id: 8, first_name: 'Rubén', last_name: 'Rivas' },
  { teacher_id: 9, first_name: 'Miguel', last_name: 'Hernández' },
  { teacher_id: 10, first_name: 'Luis', last_name: 'Gómez' },
  { teacher_id: 11, first_name: 'Andrés', last_name: 'Gálvez' },
  { teacher_id: 12, first_name: 'María', last_name: 'Fernández' },
  { teacher_id: 13, first_name: 'Elena', last_name: 'Fuertes' }
]
```