



Material complementario - Cursores

**/* Ejecución del cursor desde Node
pg-cursor */**

Ejecución del cursor desde Node pg-cursor

En pocas palabras, un cursor se puede definir como un puntero que representa una instrucción escrita con el comando SELECT y su uso radica en la mejora del rendimiento por parte del motor de base de datos, ¿Pero por qué mejora el rendimiento? Porque una vez se haya emitido una consulta con un cursor, este será reconocido en su próxima ejecución y el motor de base de datos no necesitará hacer la lectura de la sentencia completa, sino que ya sabrá exactamente qué es lo que se está solicitando.

Podrías estar haciéndote la pregunta ¿Y cómo puedo utilizar los cursores desde Node? Pues resulta que nuestro paquete protagonista “pg” es el núcleo de diferentes extensiones y paquetes creados por su autor Brian Carlson, una de estas extensiones es “pg-cursor”. Es uno de los paquetes derivados de la librería “pg”, utilizado para realizar consultas desde Node utilizando cursores.

Ejecución del cursor desde Node pg-cursor

Instalación

Como se trata de un paquete diferente deberemos instalarlo en nuestro proyecto y el comando para hacer esto es el siguiente:

```
npm i pg-cursor
```

Este paquete junto a “pg” será lo único necesario para utilizar los cursores con Node.

Sintaxis

La sintaxis de la extensión “pg-cursor” no es muy diferente a la que ya conoces del paquete “pg”, no obstante, hay que seguir unos pasos para poder emplear los cursores correctamente, desde su importación hasta su cierre.

Ejecución del cursor desde Node pg-cursor

Importación

Para hacer una consulta con cursores debes antes que todo importar el paquete “pg-cursor”, el cual devuelve por defecto una clase, así como te muestro en el siguiente código:

```
const Cursor = require("pg-cursor")
```

Consulta

Esta clase recibe como constructor un String con la consulta SQL o un JSON como argumento, así como te muestro en el siguiente código, el cuál utilizarías en caso de querer hacer una consulta parametrizada:

```
new Cursor(sql, values)
```

Ejecución del cursor desde Node pg-cursor

Consulta

De no querer parametrizar la consulta, puedes enviar como argumento al constructor la sentencia SQL en puro formato String, en el siguiente código te muestro un ejemplo de cómo sería la realización de una consulta de esta manera:

```
const consulta = new Cursor("select * from usuarios")
const cursor = client.query(consulta);
```

Observa que estoy utilizando un objeto “cliente”, el cual puede ser la instancia de la clase cliente o el parámetro que nos ofrece la conexión con la clase Pool. Si no te sientes muy ubicado en este punto no te preocupes, más adelante con un ejercicio lo verás con más claridad.

Ejecución del cursor desde Node pg-cursor

Lectura

Para leer el resultado de esta consulta, debes utilizar el método “read” de la instancia del cursor, el cual recibe como primer parámetro un valor numérico, donde se indica la cantidad de registros que el cursor debe retornar y como segundo parámetro, una función callback que recibe un posible error durante la consulta y un arreglo con los registros solicitados en formato de objeto.

En el siguiente ejemplo se muestra cómo se utiliza el método read() para leer los resultados de la consulta realizada previamente:

```
cursor.read(10, (err, rows) => {  
  console.log(rows);  
});
```

Ejecución del cursor desde Node pg-cursor

Cierre del cursor

La instancia del cursor contiene un método “close()” que al ser ejecutado, se le está indicando al motor de base de datos que una vez obtenido los registros especificados no siga con la consulta.

Este cierre se recomienda ocupar dentro del callback del método “read”, así como se muestra en

```
cursor.read(10, (err, rows) => {  
  console.log(rows);  
  cursor.close()  
});
```


/* Mi primera consulta con cursores */

Mi primera consulta con cursores

Ahora que conoces la sintaxis del paquete “pg-cursor”, procedamos con el desarrollo de una aplicación en Node, que al ser ejecutada realizará una consulta con el paquete pg-cursor a una base de datos. ¿Cuál base de datos? Revisa entre los contenidos de esta sesión y encontrarás el **Apoyo Clase - Base de datos de usuarios**. Este archivo comprimido contiene una base de datos en formato csv con 500 registros de usuarios y será la base de datos que usaremos para probar los cursores.

Para iniciar con la programación de esta aplicación deberás tener una base de datos ya creada o crear una nueva, puedes usar la siguiente instrucción para crear una base de datos llamada clientes.

Mi primera consulta con cursores

Ahora que tienes la base de datos creada, deberás crear una tabla usuarios con los campos “first_name”, “last_name” y “email”. ¿No recuerdas exactamente cómo se escribe la instrucción SQL para crear la tabla? No te preocupes, a continuación te muestro el código para hacerlo:

```
CREATE TABLE usuarios (first_name varchar(100), last_name varchar(100),  
email varchar(100));
```

Una vez creada la tabla, necesitarás importar el csv con los datos de los clientes y para eso puedes ejecutar la siguiente instrucción en la terminal de PostgreSQL.

```
\copy usuarios from '<dirección del archivo csv>' DELIMITER ',' CSV  
HEADER;
```





{desafío}
latam_

