Node.js MySQL Result Object – Examples

When a MySQL Query is executed in Node.js, an object called Result Object is returned to the callback function. The Result Object contains result set or properties that provide information regarding the execution of a query in MySQL Server.

Node.js MySQL Result Object

The contents of Result Object depends on the SQL query made to MySQL Server. Following table contents describe the result object for queries like select, insert, update and delete.

MySQL Query	Result Object
SELECT FROM	Result Set containing Record
INSERT INTO	Object containing Execution Status
UPDATE	Object containing Execution Status
DELETE FROM	Object containing Execution Status

We shall see how to access properties of records in a result set and how to access properties of execution status with the help of following examples.

- MySQL SELECT FROM Query – Accessing ResultSet
- MySQL INSERT INTO Query – Accessing properties of Result Object
- MySQL UPDATE Query -Accessing properties of Result Object
- MySQL DELETE FROM Query -Accessing properties of Result Object

Example – MySQL SELECT FROM Query – Accessing ResultSet

We can access the records in Result Set as an array and properties of a record using DOT (.) Operator.

selectUseResultObject.js - Access rows and column data of result set

```
// Node.js MySQL
Result Object Example
```

```
// Node.js MySQL Result Object Example
// include mysql module
var mysql = require('mysql');
// create a connection variable with the required details
var con = mysql.createConnection({
  host: "localhost", // ip address of server running mysql
  user: "arjun", // user name to your mysql database
  password: "password", // corresponding password
  database: "studentsDB" // use the specified database
});
// make to connection to the database.
con.connect(function(err) {
  if (err) throw err;
  // if connection is successful
  con.query("SELECT * FROM students", function (err, result, fields) {
    // if any error while executing above query, throw error
    if (err) throw err;
    // if there is no error, you have the result
    // iterate for all the rows in result
    Object.keys(result).forEach(function(key) {
      var row = result[key];
      console.log(row.name)
    });
  });
});
```

Run the above program using node in Terminal

Terminal Output

```
arjun@arjun-
VPCEH26EN:~/workspa
```

```
arjun@arjun-VPCEH26EN:~/workspace/nodejs$ node selectUseResultObject.js
John
Arjun
Prasanth
Adarsh
Raja
Sai
Ross
Monica
Lee
Bruce
Sukumar
```

Example – MySQL INSERT INTO Query

We can access the properties of a Result Object using DOT (.) Operator.

MultipleInsertExample.js - Example to access properties of result object

```
// include mysql module
var mysql =
```

```
// include mysql module
var mysql = require('mysql');
// create a connection variable with the required details
var con = mysql.createConnection({
  host: "localhost", // ip address of server running mysql
  user: "arjun", // user name to your mysql database
  password: "password", // corresponding password
  database: "studentsDB" // use the specified database
});
// make to connection to the database.
con.connect(function(err) {
  if (err) throw err;
  // if connection is successful
  var records = [
    ['Jack', 16, 82],
    ['Priya', 17, 88],
    ['Amy', 15, 74]
  ];
  con.query("INSERT INTO students (name,rollno,marks) VALUES ?", [records], function (err, result, fields) {
    // if any error while executing above query, throw error
    if (err) throw err;
    // if there is no error, you have the result
    console.log(result);
    console.log("Number of rows affected : " + result.affectedRows);
    console.log("Number of records affected with warning : " + result.warningCount);
    console.log("Message from MySQL Server : " + result.message);
  });
});
```

Run the above program using node in Terminal

Terminal Output

```
arjun@arjun-
```

```
arjun@arjun-VPCEH26EN:~/workspace/nodejs$ node MultipleInsertExample.js
OkPacket {
  fieldCount: 0,
  affectedRows: 3,
  insertId: 0,
  serverStatus: 2,
  warningCount: 0,
  message: '&Records: 3  Duplicates: 0  Warnings: 0',
  protocol41: true,
  changedRows: 0 }
Number of rows affected : 3
Number of records affected with warning : 0
Message from MySQL Server : &Records: 3 Duplicates: 0 Warnings: 0
```

Example – MySQL UPDATE Query

We can access the properties of a Result Object using DOT (.) Operator.

UpdateRecordsFiltered.js - Update records of MySQL Table

```
// include mysql module
var mysql =
```

```
// include mysql module
var mysql = require('mysql');
// create a connection variable with the required details
var con = mysql.createConnection({
  host: "localhost", // ip address of server running mysql
  user: "arjun", // user name to your mysql database
  password: "password", // corresponding password
  database: "studentsDB" // use the specified database
});
// make to connection to the database.
con.connect(function(err) {
  if (err) throw err;
  // if connection is successful
  con.query("UPDATE students SET marks=84 WHERE marks=74", function (err, result, fields) {
    // if any error while executing above query, throw error
    if (err) throw err;
    // if there is no error, you have the result
    console.log(result);
    console.log("Number of rows affected : " + result.affectedRows);
    console.log("Number of records affected with warning : " + result.warningCount);
    console.log("Message from MySQL Server : " + result.message);
  });
});
```

Run the above program using node in Terminal

Terminal Output

```
arjun@arjun-
```

```
arjun@arjun-VPCEH26EN:~/workspace/nodejs$ node UpdateRecordsFiltered.js
OkPacket {
  fieldCount: 0,
  affectedRows: 3,
  insertId: 0,
  serverStatus: 34,
  warningCount: 0,
  message: '(Rows matched: 3  Changed: 3  Warnings: 0',
  protocol41: true,
  changedRows: 3 }
Number of rows affected : 3
Number of records affected with warning : 0
Message from MySQL Server : (Rows matched: 3 Changed: 3 Warnings: 0
```

Example – MySQL DELETE FROM Query

We can access the properties of a Result Object using DOT (.) Operator.

deleteRecordsFiltered.js

```
// include mysql module
```

```
// include mysql module
var mysql = require('mysql');
// create a connection variable with the required details
var con = mysql.createConnection({
  host: "localhost", // ip address of server running mysql
  user: "arjun", // user name to your mysql database
  password: "password", // corresponding password
  database: "studentsDB" // use the specified database
});
// make connection to the database.
con.connect(function(err) {
  if (err) throw err;
  // if connection is successful
  con.query("DELETE FROM students WHERE rollno>10", function (err, result, fields) {
    // if any error while executing above query, throw error
    if (err) throw err;
    // if there is no error, you have the result
    console.log(result);
    console.log("Number of rows affected : " + result.affectedRows);
    console.log("Number of records affected with warning : " + result.warningCount);
    console.log("Message from MySQL Server : " + result.message);
  });
});
```

Run the above program using node in Terminal

Terminal Output

```
arjun@arjun-
VPCEH26EN:~/workspa
```

```
arjun@arjun-VPCEH26EN:~/workspace/nodejs$ node deleteRecordsFiltered.js
OkPacket {
  fieldCount: 0,
  affectedRows: 6,
  insertId: 0,
  serverStatus: 34,
  warningCount: 0,
  message: '',
  protocol41: true,
  changedRows: 0 }
Number of rows affected : 6
Number of records affected with warning : 0
Message from MySQL Server :
```

Conclusion

In this Node.js Tutorial – Node.js MySQL – Result Object, we have learnt to access records of a result set and also went through examples to access properties of result object containing information about query execution.

Node.js MongoDB

Node.js Mongoose

Node.js URL

Node.js FS (File System)

- Node FS - Delete a File
- Node FS Extra - Copy a Folder

Node.js JSON

- Node.js Parse JSON
- Node.js Write JSON Object to File

Node.js Error Handling

- Node.js Try Catch

Node.js Examples

- Node.js Examples
- Node.js - Handle Get Requests
- Node.js Example - Upload files to Node.js server

Useful Resources

- Node.js Interview Questions
- How to Learn Programming