



Database Connectivity

- Node.js supports all kinds of databases whether it is a relational database or NoSQL database (e.g. MongoDB)
- Download and install MySQL  
<https://dev.mysql.com/downloads/windows/installer/8.0.html>
- To access the database from Node.js, you need to install drivers for the database you want to use.

```
npm install mysql
```

```
var mysql = require('mysql');
```

Install Drivers



Create Connection



Execute SQL query

With mysql package, you may see below error -

Error: ER\_NOT\_SUPPORTED\_AUTH\_MODE: Client does not support authentication protocol requested by server; consider upgrading MySQL client

Reason: We are using mysql node package as a driver to connect to MySQL database. This package does not support new authentication method, hence MySQL db will refuse a connection.

Solution 1: Lower the security of the database by configuring it to use the old mysql\_native\_password authentication method. (*Not recommended*)

Solution 2: Use another package for mysql driver which supports the new authentication method.

```
npm install mysql2  
var mysql = require('mysql2');
```

# Create Connection



- Create connection to database, specify the username and password to connect to *MySQL* database.

```
var mysql = require('mysql2');

var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "password"
});

con.connect(function (err) {
  if (err) throw err;
  console.log("Connected!");
});
```

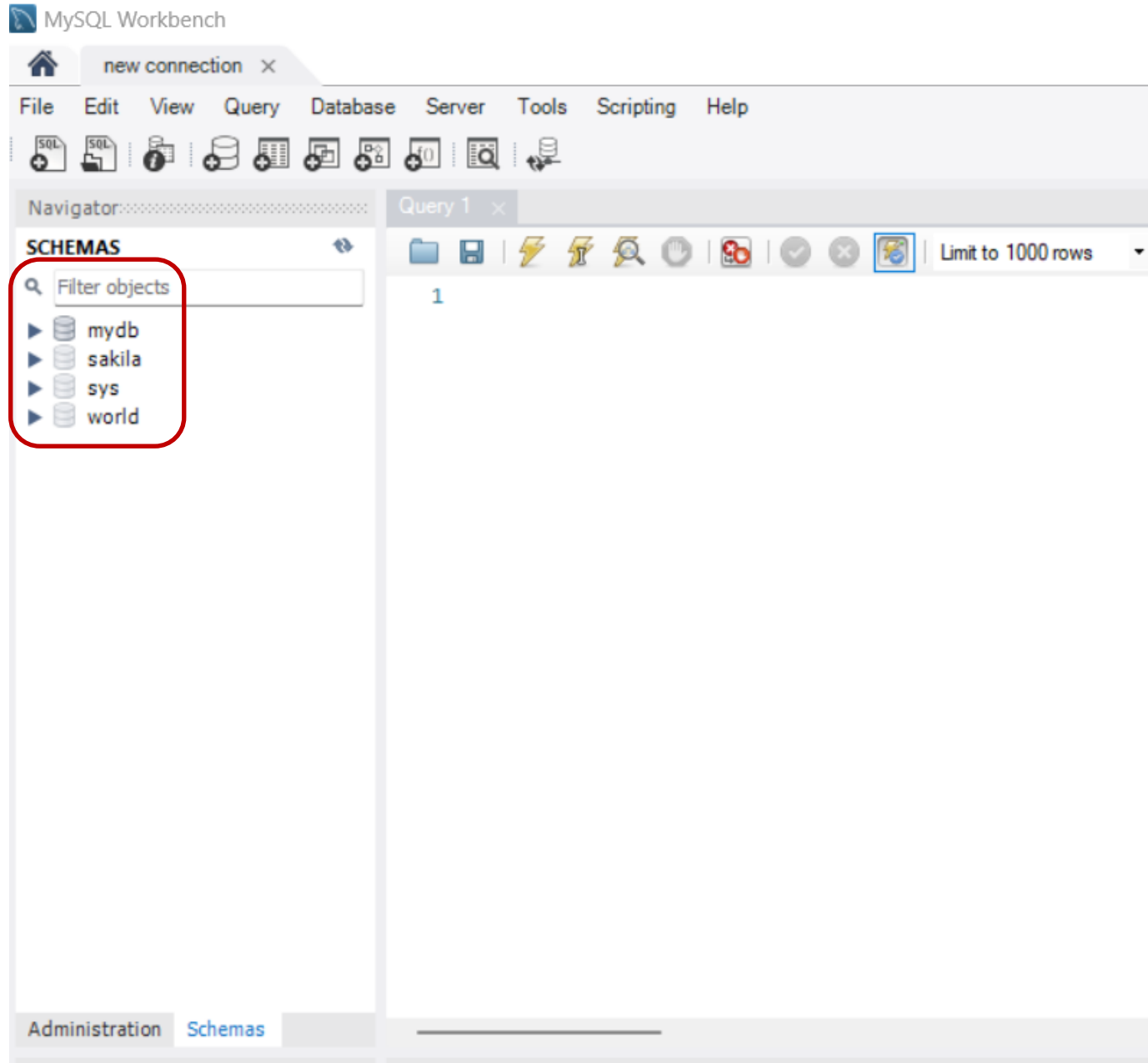
- To create a database in MySQL, use the "CREATE DATABASE" statement

```
var mysql = require('mysql2');

var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "password"
});

con.connect(function (err) {
  if (err) throw err;
  console.log("Connected!");
  con.query("CREATE DATABASE mydb", function (err, result) {
    if (err) throw err;
    console.log("Database created");
  });
});
```

# Create Database



- Never put credentials directly in the source code.
- Use environment variables to set sensitive information or use third party tools like etcd (which uses key/value pair format)

```
var mysql = require('mysql2');

var user = process.env.DB_USER;
var password = process.env.DB_PASSWORD;

var con = mysql.createConnection({
  host: "localhost",
  user: user,
  password: password
});
```

For all the programs, we will use environment variable approach to set the database credentials.



# Create Table



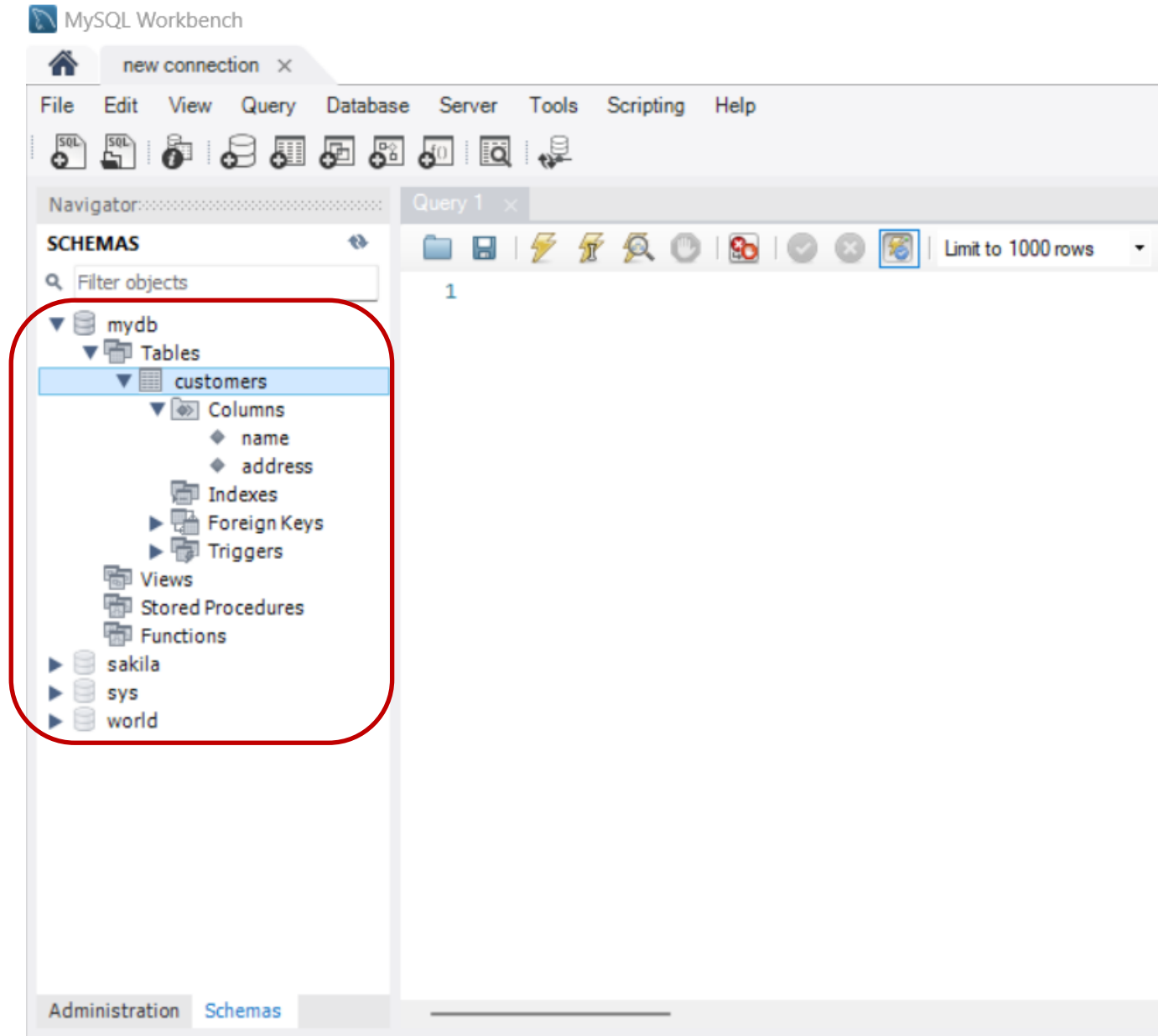
- To create a table in MySQL, use the "CREATE TABLE" statement.
- Make sure you define the name of the database when you create the connection

```
var mysql = require('mysql2');

var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "password",
  database: "mydb"
});

con.connect(function (err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "CREATE TABLE customers (name VARCHAR(255), address VARCHAR(255))";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table created");
  });
});
```

# Create Table



- To populate a table in MySQL, use the "INSERT INTO" statement

```
var mysql = require('mysql2');

var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "password",
  database: "mydb"
});

con.connect(function (err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "INSERT INTO customers (name, address) VALUES ('Google LLC', 'California')";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("1 record inserted");
  });
});
```

# Insert into table (multiple records)



- To insert more than one record, make an array containing the values, and insert a question mark (placeholder) in the sql, which will be replaced by the values array.

```
con.connect(function (err) {  
  if (err) throw err;  
  console.log("Connected!");  
  var sql = "INSERT INTO customers (name, address) VALUES ?";  
  var values = [  
    ['Google', 'California'],  
    ['Microsoft', 'Washington'],  
    ['IBM', 'New York'],  
    ['Amazon', 'Washington'],  
    ['Apple', 'California'],  
    ['Oracle', 'Texas']  
  ];  
  con.query(sql, [values], function (err, result) {  
    if (err) throw err;  
    console.log("Number of records inserted: " + result.affectedRows);  
  });  
});
```

# The Result Object



- When executing a query, a result object is returned.
- The result object contains information about how the query affected the table.

```
{  
  fieldCount: 0,  
  affectedRows: 6, // records inserted  
  insertId: 0, // id of the row just inserted  
  serverStatus: 2,  
  warningCount: 0,  
  message: '\Records:6 Duplicated: 0 Warnings: 0',  
  protocol41: true,  
  changedRows: 0  
}
```