Node.js

Lesson 10—Node.js with MongoDB and SQLite

Lesson Overview

In this lesson, you will be able to understand about the connection of Node.js with MongoDB and SQLite and perform CRUD Operations.

Learning Objectives

- ✓ Understand how Node.js connects to Database
- ✓ Define and Connect RDBMS Database and NoSQL Database
- ✓ Perform CRUD Operations

Node.js with MongoDB and SQLite

Topic 1—Node.js Connection with Database

Node.js connection with Database

All modern web applications have some type of data storage mechanism at the back end to store data.

In e-commerce applications, data such as the price of a product or the number of items of a specific type is stored in the database.

Node.js framework has a variety of options to work with databases that are commonly required by most modern web applications.

Node.js supports both relational and non-relational databases, for example, MongoDB.

Node.js connection with NoSQL Database

Node.js supports both Sqlite and MongoDB as databases. To use any of these databases, we need to download the required modules using the NPM.

For SqLite DB, the required module is “sqlite3“; for “MongoDB,” the required module is "Mongoose."

Node.js connection with NoSQL Database

You can perform the following operations in Node.js:

1. Maintain connection pooling : Developers can declare the number of Sqlite database connections that must be maintained by Node.js.
2. Create and close a connection to a database: One can mention a call-back function, which can be called whenever the "*create()*" and "*close()*" connections are triggered.
3. Queries are always executed to retrieve data from related databases.
4. Data Manipulation Language (DML) commands like insert, delete, and update can also be executed using these modules.

Node.js connection with NoSQL Database

INSTALLING THE NPM MODULES

To access MongoDB within a Node application, you need a driver. Plenty of Mongo drivers are available; however, MongoDB is among the most popular.

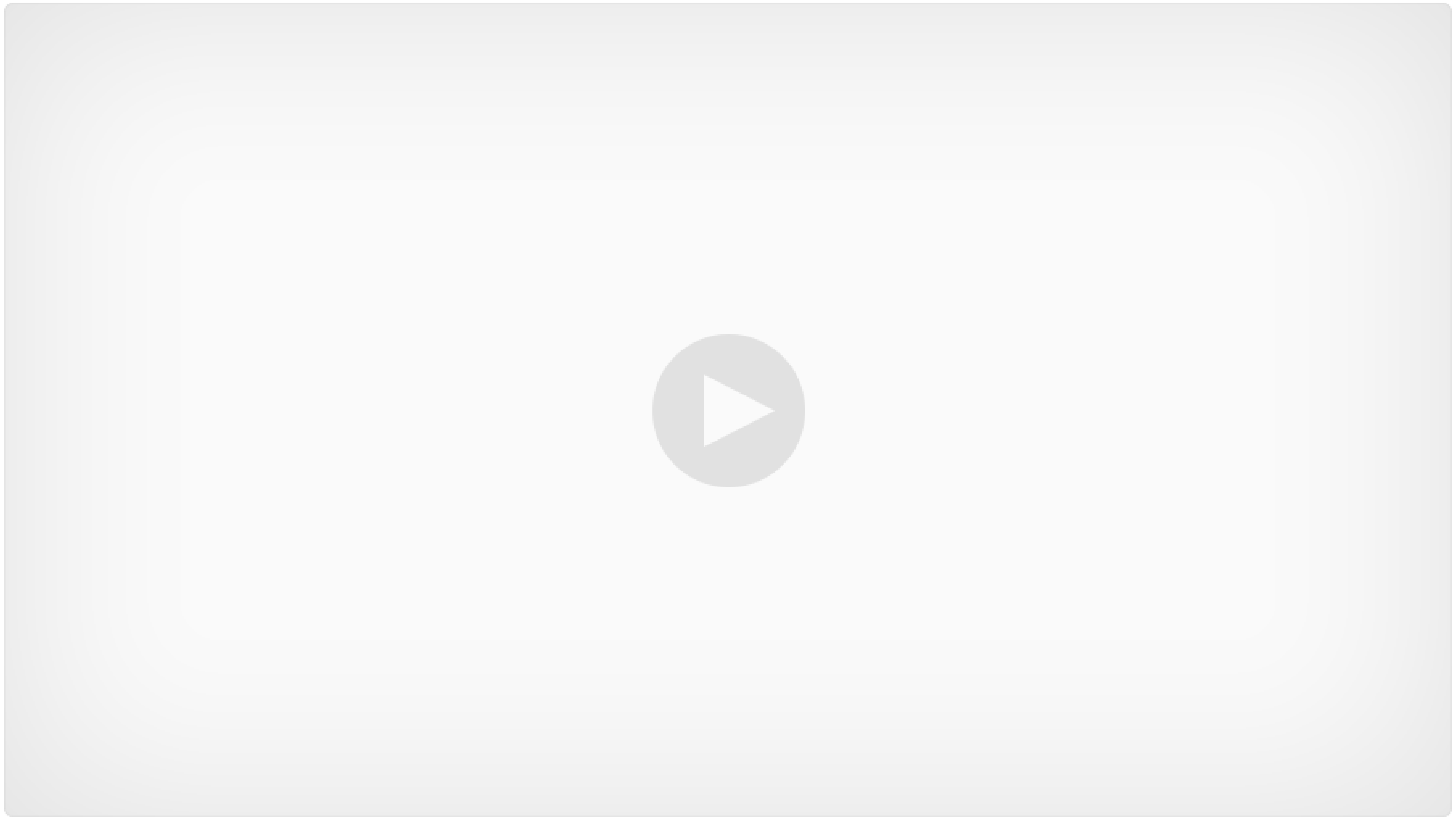
To install the MongoDB module, execute the command below:

```
npm install mongodb
```

Connect with MongoDB Database.

```
Var MongoClient = require('mongodb').MongoClient; ----------------(Using Mongod Driver)

Var url= 'mongodb://localhost:mydb;---------------- (specify the connection URL)

MongoClient.connect(url,function(err,db) {---------(Connecting to DB)

Console.log('connected'); db.close(); ------ (writing to the console log and closing the
connection)
```

Demo for Installing NPM



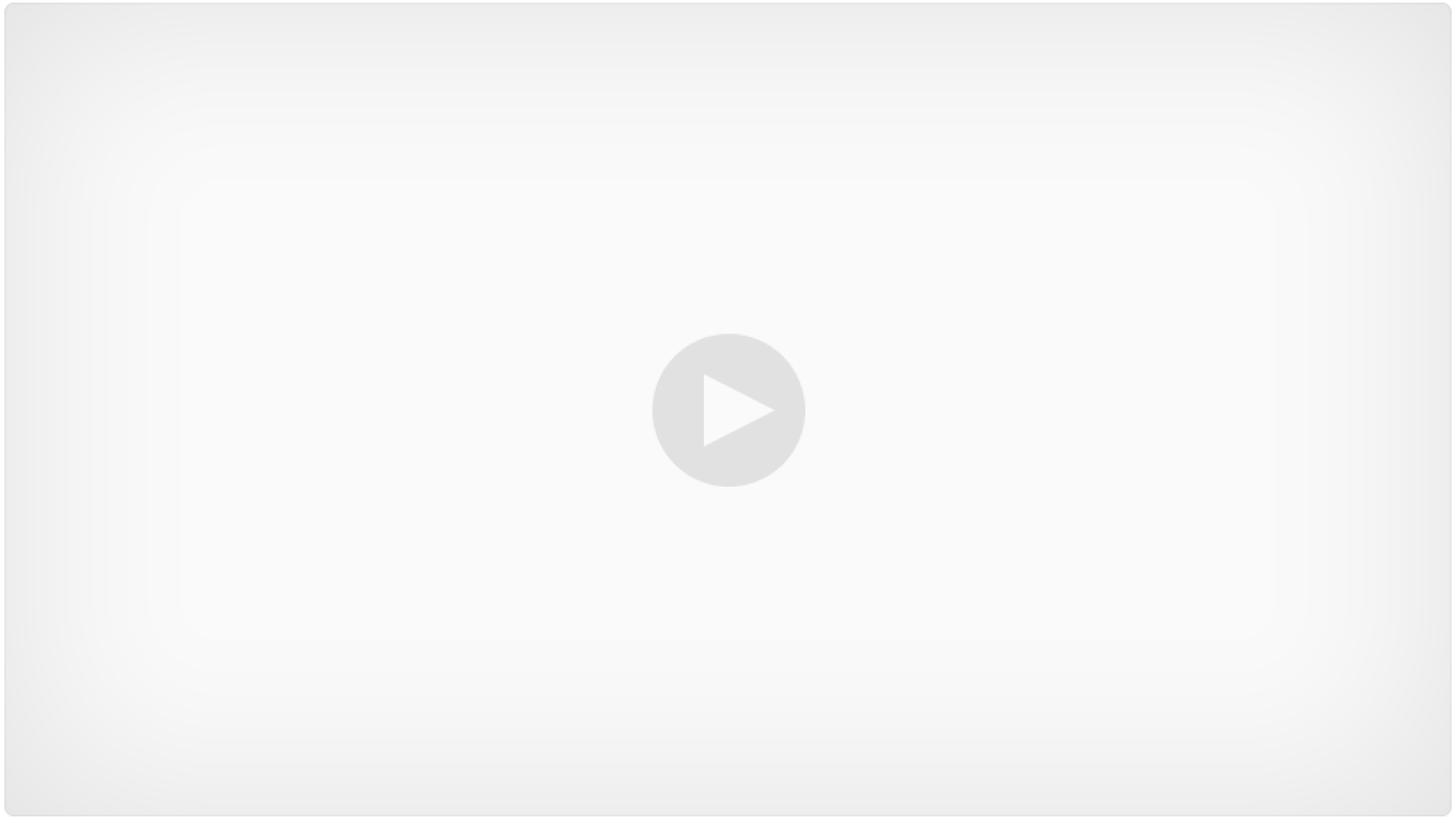
Node.js connection with NoSQL Database

QUERY FOR DATA IN MONGODB

Using MongoDB, we can apply operations to retrieve data from the MongoDB database.

```
var MongoClient = require('mongodb').MongoClient;
var url = 'mongodb://localhost/UserDetailsDB';

MongoClient.connect(url, function(err, db) {

  var cursor = db.collection(user').find();□(find() to create a cursor of record)

  cursor.each(function(err, doc) {-□ (for each record in a cursor we call function)

    console.log(doc);--------□ (printing the result)

  });
});
```

Demo for Retrieving Data from MongoDB



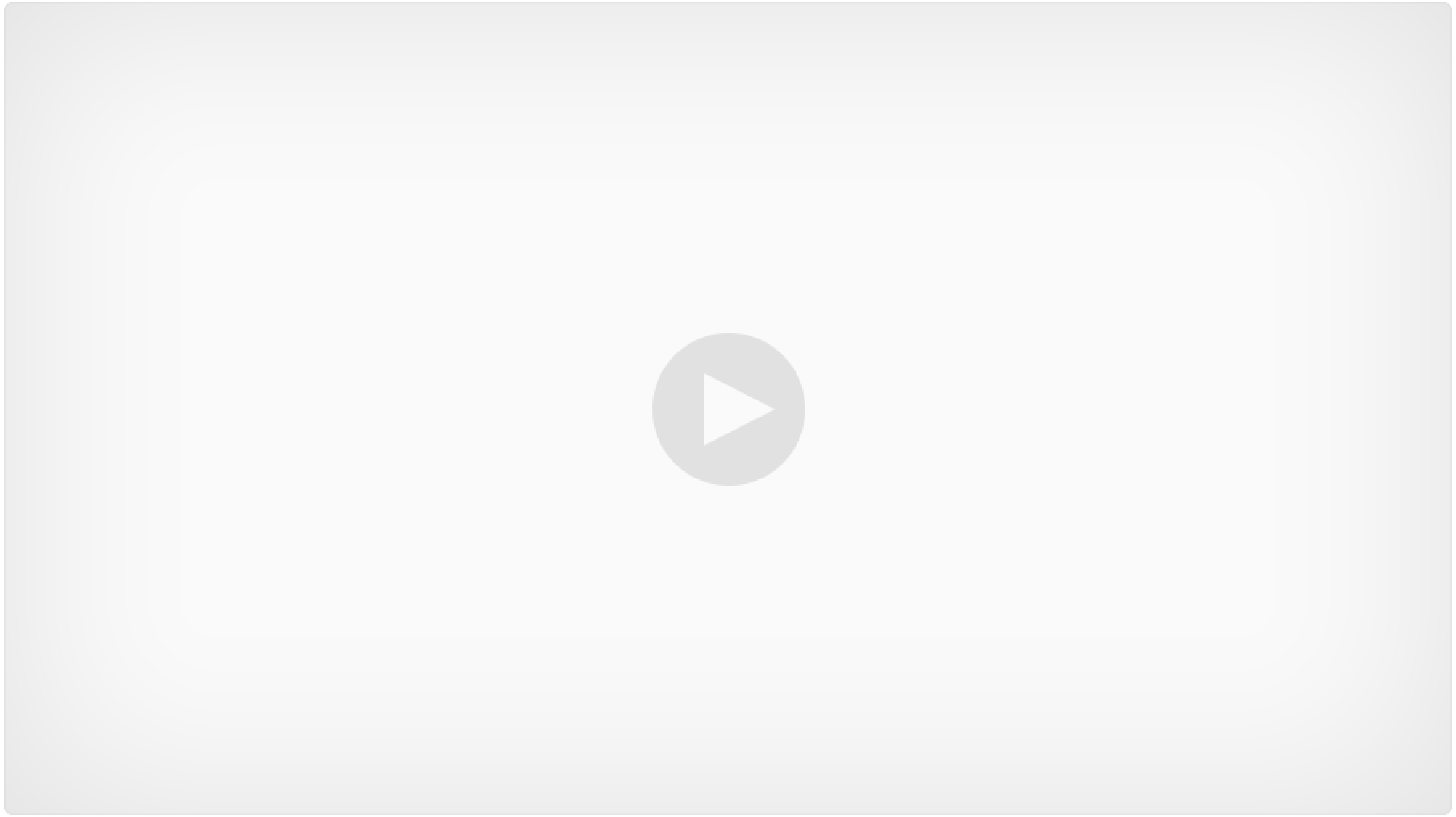
Node.js connection with NoSQL Database

INSERTING RECORD IN MONGODB

Documents can also be added into a DB collection using the method “insertOne()” offered by the MongoDB App.

```
var MongoClient = require('mongodb').MongoClient;
var url = 'mongodb://localhost/ UserDetailsDB ';

MongoClient.connect(url, function(err, db) {

  db.collection(user').insertOne({ ---□ using the insertOne to add the record
    Userid: 301,
    EmployeeName: "NewEmployee‶ -> this doc will add into the collection
  });
});
```

Demo for Inserting Data in MongoDB



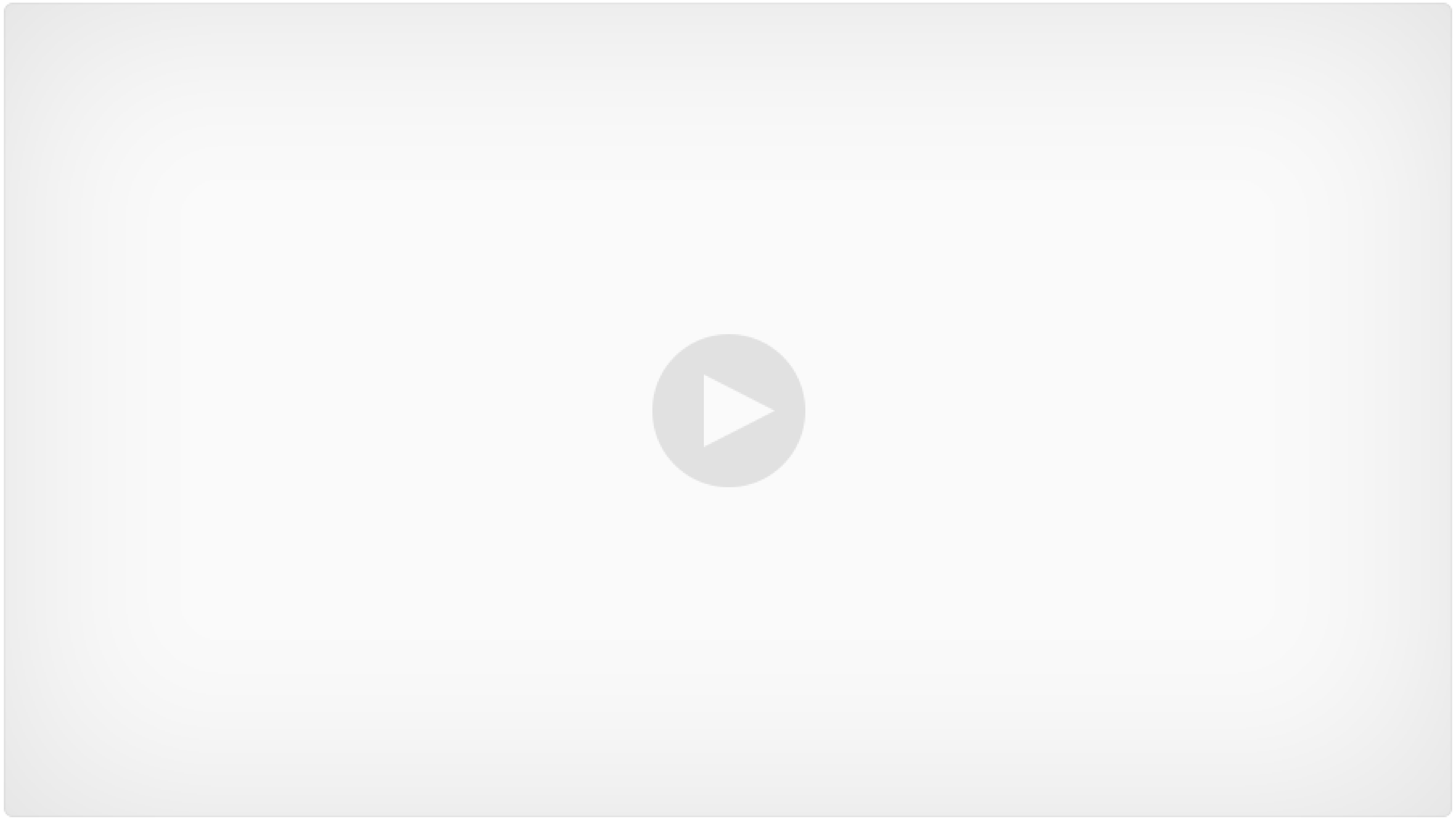
Node.js connection with NoSQL Database

UPDATING DOCUMENTS IN MONGODB

Documents can also be updated into a DB collection using the method “updateOne()” offered by the MongoDB App.

```
var MongoClient = require('mongodb').MongoClient;
var url = 'mongodb://localhost/ UserDetailsDB ';

MongoClient.connect(url, function(err, db) {

  db.collection('Employee').updateOne({ ---□ (use of updateOne Method)
    "EmployeeName": "NewEmployee‶ --□ (search Criteria for doc which need to update)
  }, {
    $set: {  ----------------□ (new value to set)
      "EmployeeName": "Mohan"
    }
  });
});
```

Demo for Updating Documents in MongoDB



Node.js connection with NoSQL Database

DELETING DOCUMENTS IN MONGODB

Documents can also be deleted from a DB collection using the method “deleteOne()” offered by the MongoDB App.

```
var MongoClient = require('mongodb').MongoClient;
var url = 'mongodb://localhost/ UserDetailsDB ';

MongoClient.connect(url, function(err, db) {

  db.collection(user').deleteOne(  -----□ (use the deleteOne Method)

    {
      "EmployeeName": “Admin“ ----(search critirea for deleting the record)
    }

  );
});
```

Demo for Deleting Documents in MongoDB

Node.js with MongoDB and SQLite

Topic 2—RDBMS Database and NoSQL Database

RDBMS Database and NoSQL Database

NODE.JS WITH RDBMS (SQLITE DB)

SQLite is a self-contained database engine that doesn't require any special server to run. This is used when the DB server is needed for other databases (For example, Oracle and MySql).

SQLite DB is one of the most popular databases for developing mobile apps and is the most widely deployed database engine in the world for portable devices.

How to install Sqlite in node.js:

```
{
 "name": "nosqlite",
 "version": "1.3.1",
 "dependencies": {
  "sqlite3": "~2.2.3"
 }
}
```

RDBMS Database and NoSQL Database

DATABASE SETUP WITH NODE.JS

```
var sql = require('sqlite3').verbose();
var db = new sqlite3.Database(‘userdb.db');
var check;
db.serialize(function() {

 db.run("CREATE TABLE if not exists Comp_INFO (info TEXT)");
 var stmt = db.prepare("INSERT INTO COMP_info VALUES (?)");
 for (var i = 0; i < 10; i++) {
   stmt.run("Ipsum " + i);
 }
 stmt.finalize();

 db.each("SELECT rowid AS id, info FROM COMP_info", function(err, row) {
   console.log(row.id + ": " + row.info);
 });
});

db.close();
```

Demo for Database Setup using Node.js

Node.js with MongoDB and SQLite

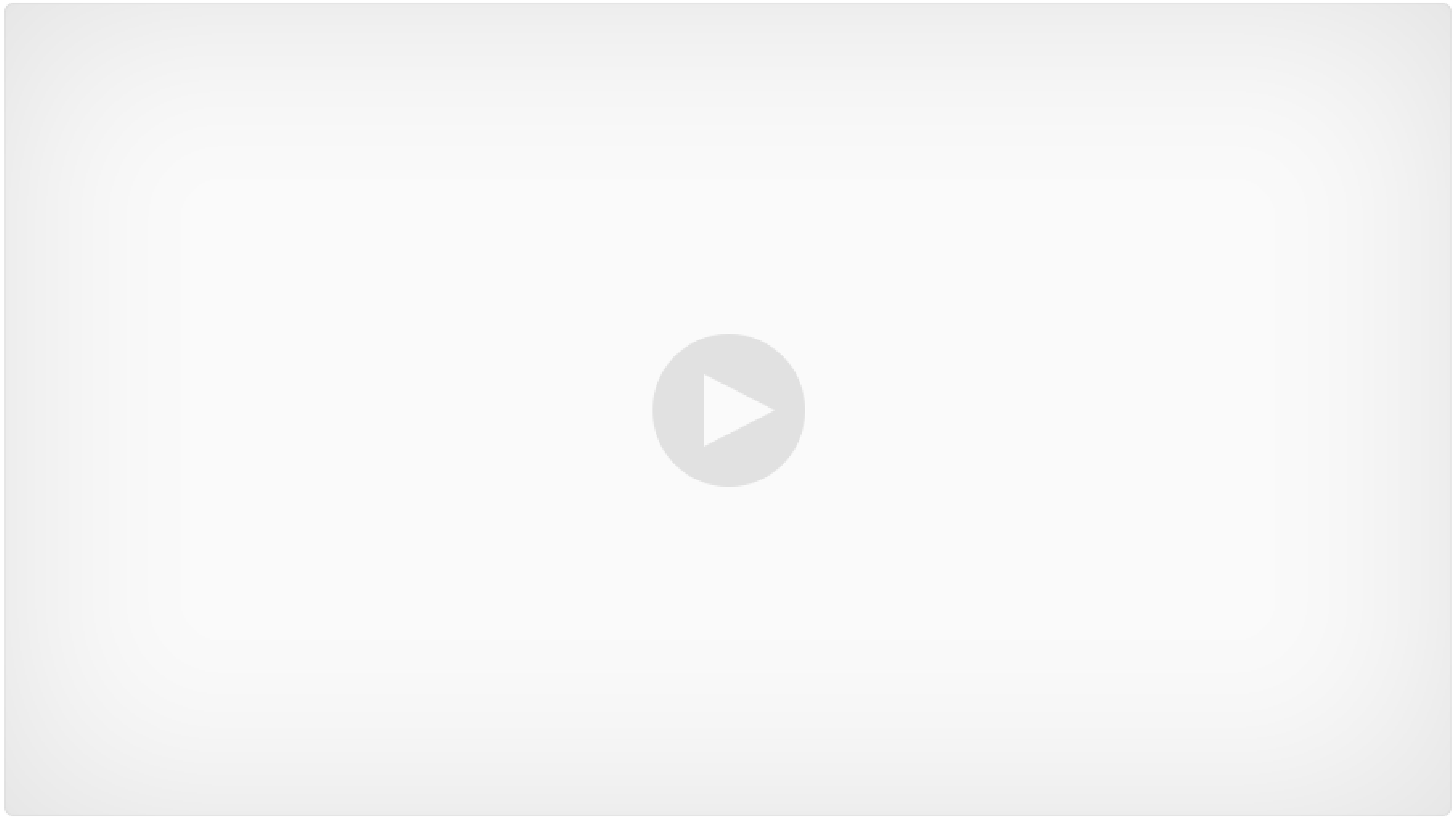
Topic 3—Performing CRUD Operations

Performing CRUD Operations

CRUD OPERATIONS WITH SQLITE

```
var sqlite3 =require('sqlite3').verbose();
var userdb = new sqlite3.Database('./db_name.db');

//Apply SELECT Operation
userdb.all("SELECT * from tabName where this="+that,function(err,rows){
//rows contain values while errors, well you can figure out.
});

//Perform INSERT operation.
db.run("INSERT into table_name(col1,col2,col3) VALUES (val1,val2,val3)");

//Perform DELETE operation
db.run("DELETE * from table_name where condition");

//Perform UPDATE operation
db.run("UPDATE table_name where condition");
```

Demo for CRUD Operations with SQLite



Quiz

QUIZ 1

MongoDB is ___ based database.

a. text

b. document

c. table

d. None of the above

QUIZ
1

MongoDB is ___ based database.

a. text

b. document

c. table

d. None of the above

The correct answer is **b. Document**

MongoDB is document-based database.

QUIZ 2

In MongoDB, “Collection” is used to describe ______.

a. database

b. table

c. tuple

d. column

QUIZ

2

In MongoDB, “Collection” is used to describe ______.

a. database

b. table

c. tuple

d. column

The correct answer is **b. table.**

In MongoDB, “Collection” is used to describe table.

QUIZ

3

Sequelize is used as “____” tool.

a. ORM

b. database

c. relationship

d. All of the above

QUIZ 3

Sequelize is used as "_____" tool.

a. ORM

b. database

c. relationship

d. All of the above

The correct answer is **a. ORM.**

Sequelize is used as "ORM" tool.

QUIZ

4

Which function is used to delete the single record in MongoDB?

a. deleteSingle

b. removeOne

c. deleteOne

d. remove

QUIZ 4

Which function is used to delete the single record in MongoDB?

a. deleteSingle

b. removeOne

c. deleteOne

d. remove

The correct answer is **c. deleteOne.**

---

deleteOne function is used to delete the single record in MongoDB.

Key Takeaways

- ✓ Node.js supports both Sqlite and MongoDB. To use these databases, we need to download the required modules using the NPM.
- ✓ SQLite is a self-contained database engine that doesn't require any special server to run. This is used when the DB server is needed for other databases (For example, Oracle and MySql).
- ✓ There is a script for performing CRUD Operations with SQLite.

simplilearn

Thank You