# NODE JS APPLICATION WITH SQL DATABASE

GROUP-KERALA CODERS

October 2019

Group Members

Bincy Alias, Cherin Linson, Jestin Skaria, Keerthi Raveendran

prezi presentation:
https://prezi.com/p/p$_{h}nl9jctxr - /$

## 1 NODE

- Open-source server side run-time environment
- Built on Chrome's V8 JavaScript engine.
- Uses JavaScript to build entire server side application.
- Asynchronous by default

## 2 NODE JS WITH SQLITE

- SQLite an open-source, embedded, relational database management system
- SQLite is not a client-server DBMS
- In-memory library that you can call and use directly
- How to interact with SQLite databases using Node JS
- application using the sqlite3 module?

## 3 Steps for Connecting to SQLite database

- Install sqlite3 module
- In command prompt, type following
- npm install sqlite3
- To connect to new database using node js
- const sqlite3 = require('sqlite3');
- To transfer data over the Hyper Text Transfer Protocol
- var http = require('http');
- require() method: to include modules
- HTTP module - create an HTTP server -¿listens to server
- ports-¿client will get response.

## 4 Code to create tables

- Our application is having a database
- studentEnrollmentSystem With 3 tables
- =¿students, classes and enrollment

## 5 Data for tables

- var studentData = ["studentId":"1","studentName":"Jestin","programGroup":"CPCT", "studentId":"2","studentName":"Cherin","programGroup":"CPCT", "studentId":"3","studentName":"Bincy","programGroup":"CPCT", "studentId":"4","studentName":"Keerthy","programGroup":"CPCT", "studentId":"5","studentName":"Susan","programGroup":"CSAT", "studentId":"6","studentName":"Diana","programGroup":"CSAT"];

## 6 Serialize function and inserting data

- db.serialize(function() studentData.forEach(function(studentData) db.run('INSERT INTO students(studentid,studentname,programgroup) VALUES(?,?,?)',[studentData.studentId,stude function(err) if (err) return console.log(err.message); console.log('A row has been inserted with rowid *this.lastID*');
- The serialize() method-to make excecution in a serialized manner, means that only one statement can execute at a time.

- Other statements-wait in a queue until all the previous statements are executed.
- Here data from JSON array is taken like studentData.studentId Similarly, we enter data for other 2 tables in same manner.

## 7 Selecting and displaying data with the help of Serialize()

- db.serialize(function()

  var data = []; db.all(”SELECT s.studentid,s.studentname,c.classid,c.classname,c.room,c.datesession FROM students AS s INNER JOIN enrollment as e on e.studentid=s.studentid INNER JOIN classes as c on c.classid = e.classid”, function(err,rows) if(err) return console.log(err.message); ; console.log(”================STUDENT SCHEDULE DETAILS============”); console.log(”STUDENT ID: STUDENT NAME : CLASSID: CLASS NAME : ROOM : DATE SESSION ============”); rows.forEach(function (row) console.log( row.studentid + ” ; ” + row.studentname + ” ; ” + row.classid + ” ; ” + row.classname + ” ; ” + row.room + ” ; ” + row.datesession ); ); ); ); );

## 8 Selecting and displaying data with the help of Serialize()

- In above code we display content of all rows in table using serialize()
- Select query
- forEach()

## 9 Code

- var bodyParser = require(’body-parser’); const sqlite3 = require(’sqlite3’) var http = require(’http’); var urlencodedParser = bodyParser.urlencoded( extended: true ); let db = new sqlite3.Database(”./a3”, (err) =¿ if (err) console.log(’Error when creating the database’, err) else

  var sql1 = db.run(”CREATE TABLE IF NOT EXISTS students (studentid VARCHAR(20), studentname VARCHAR(20) , programgroup VARCHAR(20))”);

  var sql2 = db.run(”CREATE TABLE IF NOT EXISTS classes (classid VARCHAR(20), classname VARCHAR(255) , room VARCHAR(25),datesession VARCHAR(20))”);

var sql3 = db.run(”CREATE TABLE IF NOT EXISTS enrollment (enrollmentid VARCHAR(20), studentid VARCHAR(20) , classid VARCHAR(25))”);

var studentData = [”studentId”:”1”,”studentName”:”Jestin”,”programGroup”:”CPCT”, ”studentId”:”2”,”studentName”:”Cherin”,”programGroup”:”CPCT”, ”studentId”:”3”,”studentName”:”Bincy”,”programGroup”:”CPCT”, ”studentId”:”4”,”studentName”:”Keerthy”,”programGroup”:”CPCT”, ”studentId”:”5”,”studentName”:”Susan”,”programGroup”:”CSAT”, ”studentId”:”6”,”studentName”:”Diana”,”programGroup”:”CSAT”];

db.serialize(function() studentData.forEach(function(studentData) /* inserting data*/ db.run(‘INSERT INTO students(studentid,studentname,programgroup) VALUES(?,?,?)‘,[studentData.studentId,studentData.studentName,studentData.programGroup], function(err) if (err) return console.log(err.message); // get the last insert id console.log(‘A row has been inserted with rowid *this.lastID*‘); ); ); ); *varclassSection* = [”*classId*” : ”*C*104”, ”*classNmae*” : ”*JAVA*”, ”*room*” : ”*B*230”, ”*dateSession*” : ”*THUAM*”, ”*class*

db.serialize(function() classSection.forEach(function(classSection) db.run(‘INSERT INTO classes(classid,classname,room,datesession) VALUES(?,?,?,?)‘, [classSection.classId,classSection.classNmae,classSection.room,classSection.dateSession], function(err) if (err) return console.log(err.message); // get the last insert id console.log(‘A row has been inserted with rowid *this.lastID*‘); ); ); );

var enroll =[”enrollmentId”:”1”,”studentId”:”1”,”classId”:”C104”, ”enrollmentId”:”2”,”studentId”:”2”,”classId”:”C104”, ”enrollmentId”:”3”,”studentId”:”3”,”classId”:”C
”enrollmentId”:”4”,”studentId”:”1”,”classId”:”C105”, ”enrollmentId”:”5”,”studentId”:”2”,”classId”:
”enrollmentId”:”6”,”studentId”:”3”,”classId”:”C105”, ”enrollmentId”:”7”,”studentId”:”4”,”classId”:
”enrollmentId”:”8”,”studentId”:”5”,”classId”:”C109”, ”enrollmentId”:”9”,”studentId”:”6”,”classId”:
”enrollmentId”:”10”,”studentId”:”5”,”classId”:”C110”, ”enrollmentId”:”11”,”studentId”:”6”,”classId”:”C109”]; db.serialize(function() enroll.forEach(function(enroll) db.run(‘INSERT INTO enrollment(enrollmentid,studentid,classid) VALUES(?,?,?)‘, [enroll.enrollmentId,enroll.studentId,enroll.classId], function(err) if (err) return console.log(err.message); // get the last insert id console.log(‘A row has been inserted with rowid *this.lastID*‘); ); ); );

db.serialize(function()

var data = []; db.all(”SELECT s.studentid,s.studentname,c.classid,c.classname,c.room,c.datesession FROM students AS s INNER JOIN enrollment as e on e.studentid=s.studentid INNER JOIN classes as c on c.classid = e.classid”, function(err,rows) if(err) return console.log(err.message); ;

console.log(”=================STUDENT SCHEDULE DETAILS============”);

console.log(”STUDENT ID: STUDENT NAME : CLASSID: CLASS NAME : ROOM : DATE SESSION ============”); rows.forEach(function (row) console.log( row.studentid + ” ; ” + row.studentname + ” ; ” + row.classid + ” ; ” + row.classname + ” ; ” + row.room + ” ; ” + row.datesession ); );

); ); )