Assignment 7

Josh Valentino

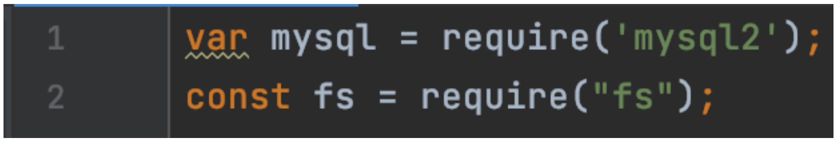
CSC 4500

Professor Casey Walters

11/28/2021

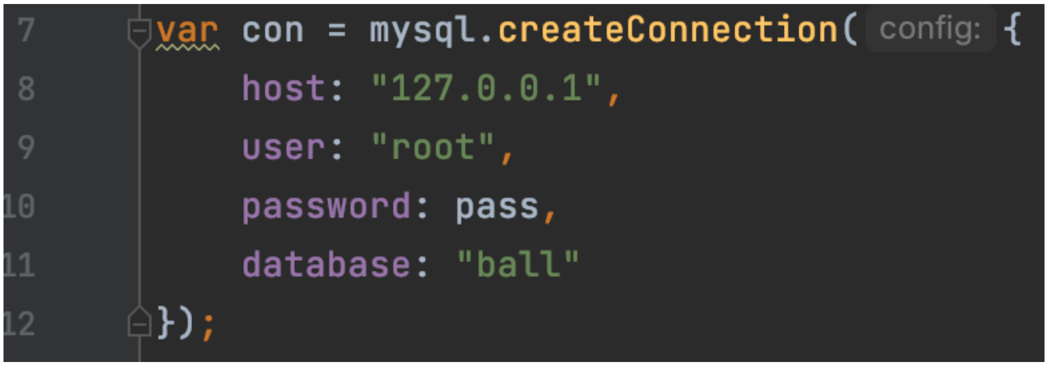
Assignment 7

In order to connect to the database via web application, node.js was used. The modules mysql2 and fs were utilized. The figure below shows the code for both the modules used.



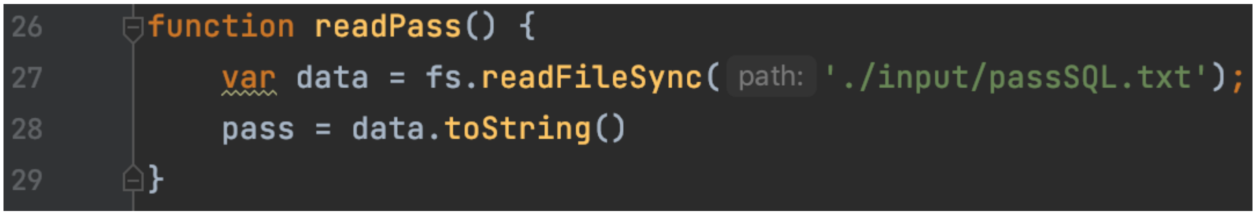
```
var mysql = require('mysql2');
const fs = require("fs");
```

An SQL connection was created with the host, user, and database being hardcoded. The figure below shows the code for the creation of the SQL connection.



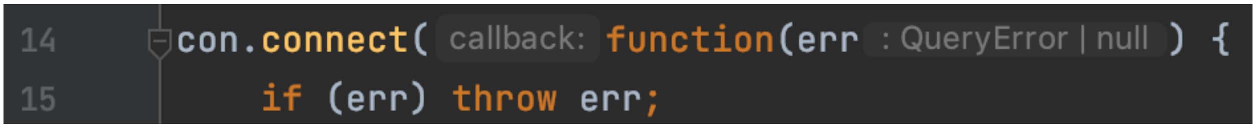
```
var con = mysql.createConnection( config: {
    host: "127.0.0.1",
    user: "root",
    password: pass,
    database: "ball"
});
```

The password was written into a text file and is obtained using file system (fs) read file sync. While this is not inherently secure, it does provide the ability to record a Flipgrid without all the viewers seeing the password directly. The figure below shows the code for function readPass, which utilizes the fs module to read in the password.



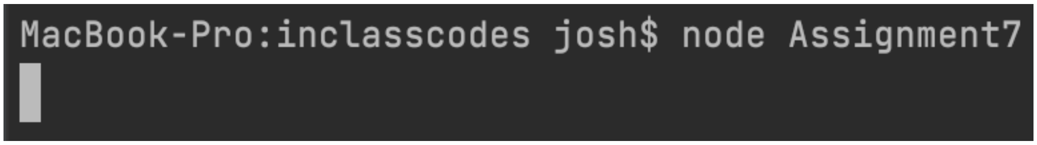
```
function readPass() {
    var data = fs.readFileSync( path: './input/passSQL.txt');
    pass = data.toString()
}
```

The connection is confirmed due to the lack of error that is thrown when running. The figure below shows the code for the connection attempt. It shows the if statement that would trigger an error to be thrown if there was an error connecting.



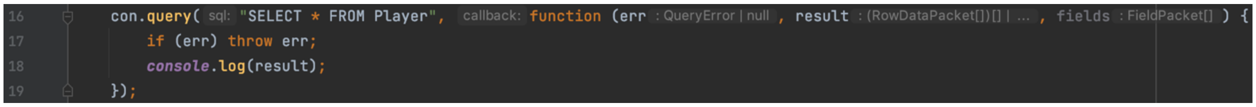
```
con.connect( callback: function(err : QueryError | null ) {
        if (err) throw err;
```

The figure below shows the lack of error being thrown after the JavaScript file was ran with node, confirming successful connection.



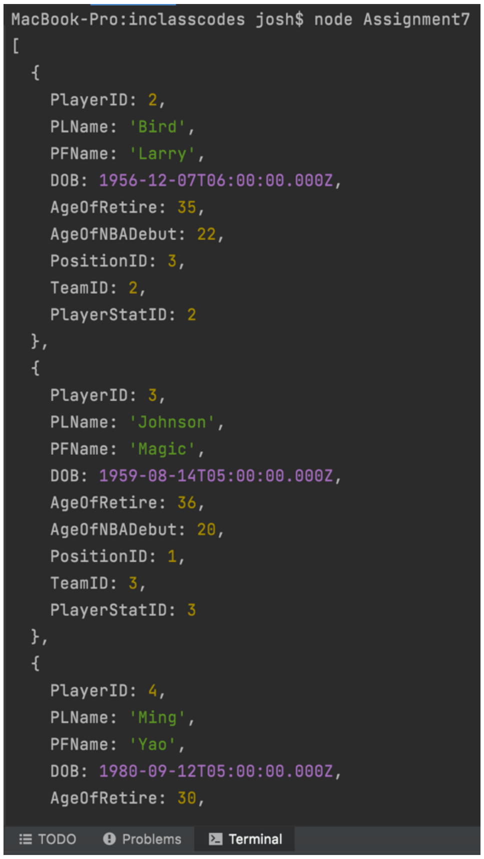
```
MacBook-Pro:inclasscodes josh$ node Assignment7
```

Two queries are ran to further prove the connection status. The figure below shows the code for the simple query ran. It utilizes the query function of the connection variable, con.



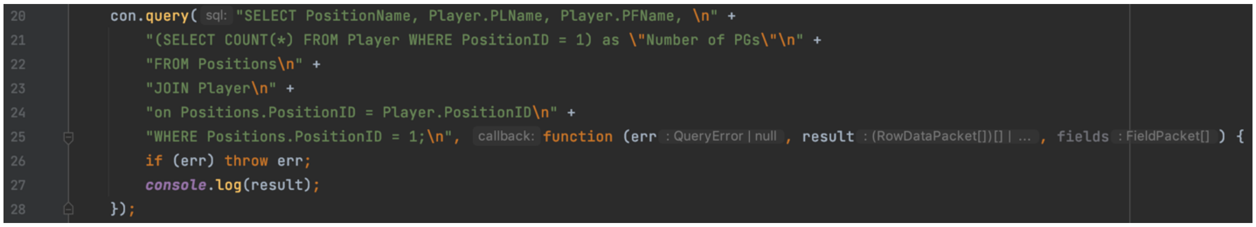
```
con.query( sql: "SELECT * FROM Player",  callback: function (err : QueryError | null , result : (RowDataPacket[])[] | … , fields : FieldPacket[] ) {
    if (err) throw err;
    console.log(result);
});
```

The figure below shows part of the result of the simple query.



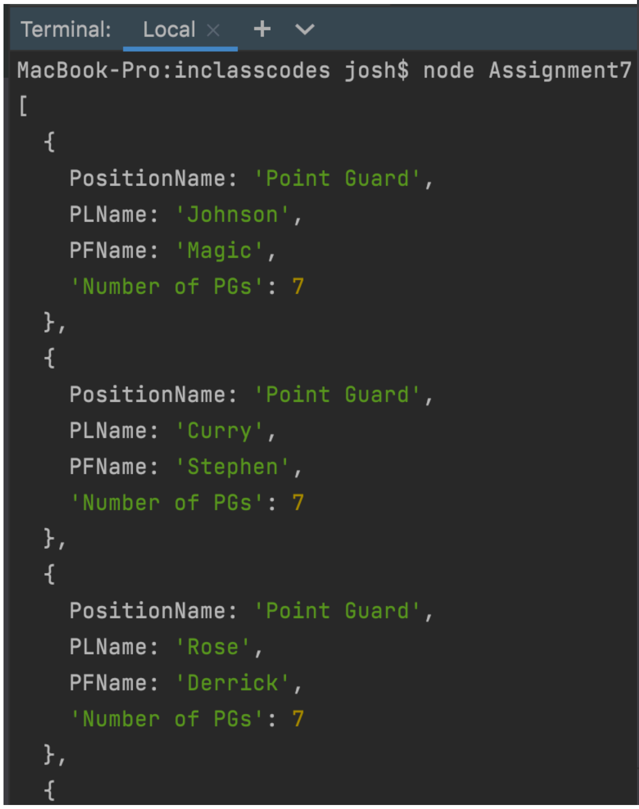
```
MacBook-Pro:inclasscodes josh$ node Assignment7
[
  {
    PlayerID: 2,
    PLName: 'Bird',
    PFName: 'Larry',
    DOB: 1956-12-07T06:00:00.000Z,
    AgeOfRetire: 35,
    AgeOfNBADebut: 22,
    PositionID: 3,
    TeamID: 2,
    PlayerStatID: 2
  },
  {
    PlayerID: 3,
    PLName: 'Johnson',
    PFName: 'Magic',
    DOB: 1959-08-14T05:00:00.000Z,
    AgeOfRetire: 36,
    AgeOfNBADebut: 20,
    PositionID: 1,
    TeamID: 3,
    PlayerStatID: 3
  },
  {
    PlayerID: 4,
    PLName: 'Ming',
    PFName: 'Yao',
    DOB: 1980-09-12T05:00:00.000Z,
    AgeOfRetire: 30,
```

The figure below shows the code for the complex query ran. It utilizes the query function of the connection variable, con.

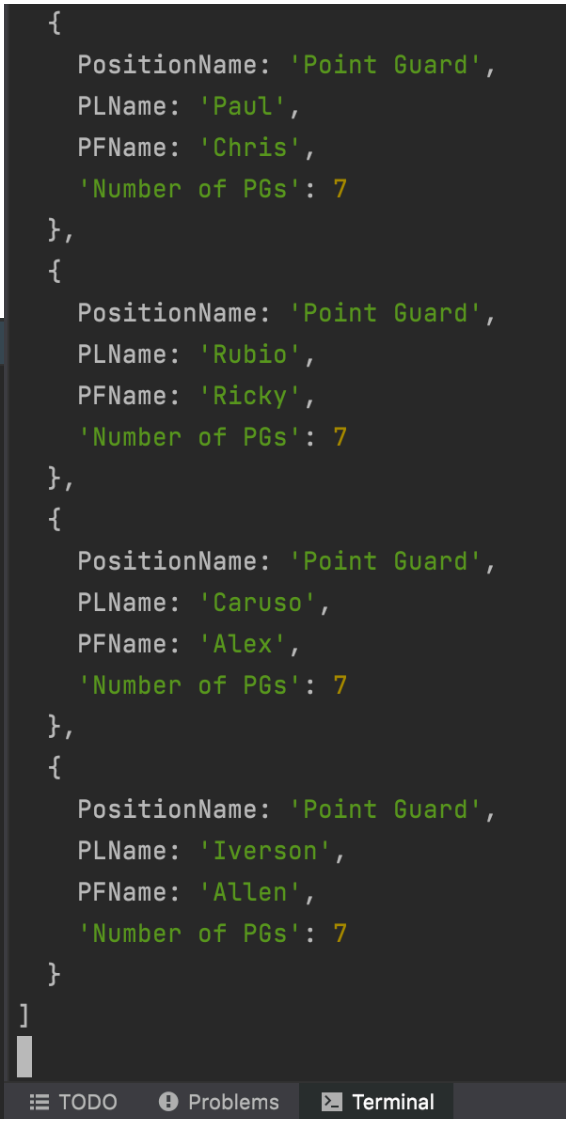


```
    con.query( sql: "SELECT PositionName, Player.PLName, Player.PFName, \n" +
        "(SELECT COUNT(*) FROM Player WHERE PositionID = 1) as \"Number of PGs\"\n" +
        "FROM Positions\n" +
        "JOIN Player\n" +
        "on Positions.PositionID = Player.PositionID\n" +
        "WHERE Positions.PositionID = 1;\n",  callback: function (err : QueryError | null , result : (RowDataPacket[])[] | ... , fields : FieldPacket[] ) {
        if (err) throw err;
        console.log(result);
    });
```

The figures below show the result of the complex query.



```
MacBook-Pro:inclasscodes josh$ node Assignment7
[
  {
    PositionName: 'Point Guard',
    PLName: 'Johnson',
    PFName: 'Magic',
    'Number of PGs': 7
  },
  {
    PositionName: 'Point Guard',
    PLName: 'Curry',
    PFName: 'Stephen',
    'Number of PGs': 7
  },
  {
    PositionName: 'Point Guard',
    PLName: 'Rose',
    PFName: 'Derrick',
    'Number of PGs': 7
  },
  {
```



```
  {
    PositionName: 'Point Guard',
    PLName: 'Paul',
    PFName: 'Chris',
    'Number of PGs': 7
  },
  {
    PositionName: 'Point Guard',
    PLName: 'Rubio',
    PFName: 'Ricky',
    'Number of PGs': 7
  },
  {
    PositionName: 'Point Guard',
    PLName: 'Caruso',
    PFName: 'Alex',
    'Number of PGs': 7
  },
  {
    PositionName: 'Point Guard',
    PLName: 'Iverson',
    PFName: 'Allen',
    'Number of PGs': 7
  }
]
```