# Node, the SQL

# Objectives

Today we return to JavaScript with our newly acquired SQL skills and combine them to create back-end applications that utilize MySQL data. In this way, you will learn how to manipulate back-end data through applications.

- To create a connection to a MySQL database using Node
- To create, read, update, and delete data from a MySQL database using Node
- To work with a group in taking a basic concept for a server side application and creating a working prototype for that application within a given time frame

# Demo: Creating a Database Connection

$ node iceCreamBasic.js

npm install mysql

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

```
var connection = mysql.createConnection({
  host: "localhost",
  port: 3306,

  // Your username
  user: 'root',

  // Your password
  password: '',
  database: 'ice_creamDB'
});
```

What’s interesting about this?

We’re passing our MySQL connection to a variable

# Now let’s connect Node to MySQL

```
connection.connect(function(err){
  if(err) throw err;
  console.log("connected as id "+ connection.threadId);
});
```

RTFM https://www.npmjs.com/package/mysql

# Activity: Creating a Database Connection (15 min)

# Demo: Reading from a Database

07-iceCreamReadData

$ node iceCreamDBConnections.js

```
connection.query("SELECT * FROM products", function(err, res) {
    if (err) throw err;
    console.log(res);
  });
```

connection.query(<STRING>,function(err, res) {})

Be very careful with syntax!
Even small differences can lead to an error being returned.

# RowDataPacket?

# Activity: Collecting Data from a Database (20 min)

See 08-playlistRead for Instructions

"SELECT * FROM songs WHERE genre=?", ["Dance"],

This allows us to place an array after the query string whose contents will replace the question marks with those variables contained within the array.

# What is CRUD?

# The Four Basic Functions of Persistent Storage

- C - CREATE - INSERT INTO pets (name, type, age) VALUES ("fido", "dog", 3);
- R - READ  - SELECT * FROM pets;
- U - UPDATE - UPDATE pets SET name="under dog" WHERE type = "dog";
- D - DELETE - DELETE FROM pets WHERE type = "mouse";

RTFM https://en.wikipedia.org/wiki/Create,_read,_update_and_delete

# Demo: 09-iceCreamCRUD

This looks similar to the code which reads data, with the only major differences being the query made and the data entered

```
var query = connection.query(

console.log(query.sql);
```

When inserting data into a MySQL database using Node, the format is to use object notation with the keys being the columns that you would like to insert data into

"UPDATE products SET ? WHERE ?",

What’s interesting about this?

By using an array, we are able to replace both question marks with the elements contained within

# Activity: CRUD Playlist (30 min)

# Activity: Share Your Work

# Activity: Great Bay (90 min)

# Activity: Share Your Work (30 min)

# Demo: Homework