**Part A (To be referred by students)**

***Forms, Input Types & Media***

**SAVE THE FILE AND UPLOAD AS (RollNo\_Name\_Exp2)**

**Topic covered:** **server-side database connectivity using Node.js**

**Learning Objective: Learner would be able to**

1. To present an introduction to NodeJs and Mysql Database to understand the backend database connection through scripting language.
2. MySql is one of the most popular database language used for storing an application data.

**Aim:**

**To learn and implement server-side database connectivity (Mysql ) using Node.js .**

**Prerequisites:-**

* Basic NodeJs and database query language operation

**Outcomes:-**

* Students will get idea about how to work with Server side database.
* Students will be able to work with different database operation like insert, select , update and delete.

**Node.js MySQL**

**MySQL Database**

To be able to experiment with the code examples, you should have MySQL installed on your computer.

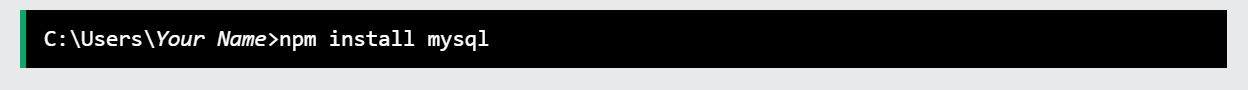
**Install MySQL Driver**

Once you have MySQL up and running on your computer, you can access it by using Node.js.

To access a MySQL database with Node.js, you need a MySQL driver. This tutorial will use the "mysql" module, downloaded from NPM.

To download and install the "mysql" module, open the Command Terminal and execute the following:

C:\Users\Your Name>npm install mysql



Now you have downloaded and installed a mysql database driver.

Node.js can use this module to manipulate the MySQL database:

var mysql = require('mysql');

Create Connection

Start by creating a connection to the database.

Use the username and password from your MySQL database.

**demo\_db\_connection.js**

var mysql = require('mysql');

var con = mysql.createConnection({

host: "localhost",

user: "yourusername",

password: "yourpassword"

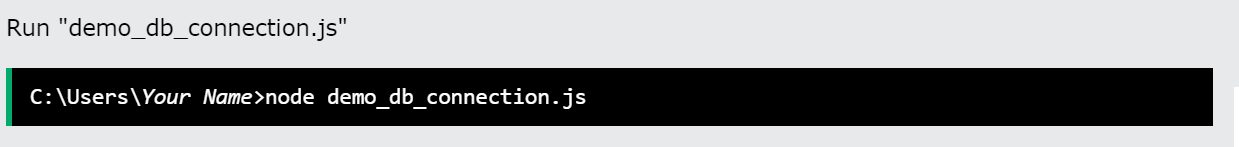
});

con.connect(function(err) {

if (err) throw err;

console.log("Connected!");});

Save the code above in a file called "demo\_db\_connection.js" and run the file:



Which will give you this result:



**Query a Database**

Use SQL statements to read from (or write to) a MySQL database. This is also called "to query" the database.

The connection object created in the example above, has a method for querying the database:

con.connect(function(err) {

if (err) throw err;

console.log("Connected!");

con.query(sql, function (err, result) {

if (err) throw err;

console.log("Result: " + result);

});

});

The query method takes an sql statements as a parameter and returns the result.

Learn how to read, write, delete, and update a database in the next chapters.

**Node.js MySQL Create Database**

**Creating a Database**

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

Create a database named "mydb":

var mysql = require('mysql');

var con = mysql.createConnection({

host: "localhost",

user: "yourusername",

password: "yourpassword"

});

con.connect(function(err) {

if (err) throw err;

console.log("Connected!");

con.query("CREATE DATABASE mydb", function (err, result) {

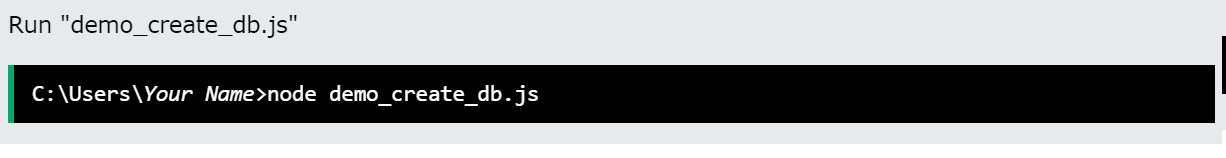
if (err) throw err;

console.log("Database created");

});

});

Save the code above in a file called "demo\_create\_db.js" and run the file:



Which will give you this result:



**Creating a 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('mysql');

var con = mysql.createConnection({

host: "localhost",

user: "yourusername",

password: "yourpassword",

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");

});

});

**Insert Into Table**

Insert a record in the "customers" table:

var mysql = require('mysql');

var con = mysql.createConnection({

host: "localhost",

user: "yourusername",

password: "yourpassword",

database: "mydb"

});

con.connect(function(err) {

if (err) throw err;

console.log("Connected!");

var sql = "INSERT INTO customers (name, address) VALUES ('Company Inc', 'Highway 37')";

con.query(sql, function (err, result) {

if (err) throw err;

console.log("1 record inserted");

});

});

**Selecting From a Table**

To select data from a table in MySQL, use the "SELECT" statement.

var mysql = require('mysql');

var con = mysql.createConnection({

host: "localhost",

user: "yourusername",

password: "yourpassword",

database: "mydb"

});

con.connect(function(err) {

if (err) throw err;

con.query("SELECT \* FROM customers", function (err, result, fields) {

if (err) throw err;

console.log(result);

});

});

**Select With a Filter:**

"SELECT \* FROM customers WHERE address = 'Park Lane 38'"

**Sort the Result**

Use the ORDER BY statement to sort the result in ascending or descending order.

"SELECT \* FROM customers ORDER BY name"

**Delete Record**

var sql = "DELETE FROM customers WHERE address = 'Mountain 21'";

**Delete a Table**

You can delete an existing table by using the "DROP TABLE" statement:

var sql = "DROP TABLE customers";

**Update Table**

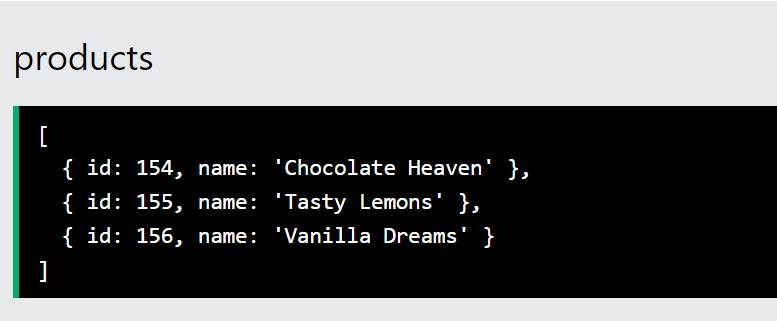
var sql = "UPDATE customers SET address = 'Canyon 123' WHERE address = 'Valley 345'";

**Join Two or More Tables**

You can combine rows from two or more tables, based on a related column between them, by using a JOIN statement.

Consider you have a "users" table and a "products" table:





These two tables can be combined by using users' favorite\_product field and products' id field.

var mysql = require('mysql');

var con = mysql.createConnection({

host: "localhost",

user: "yourusername",

password: "yourpassword",

database: "mydb"

});

con.connect(function(err) {

if (err) throw err;

var sql = "SELECT users.name AS user, products.name AS favorite FROM users JOIN products ON users.favorite\_product = products.id";

con.query(sql, function (err, result) {

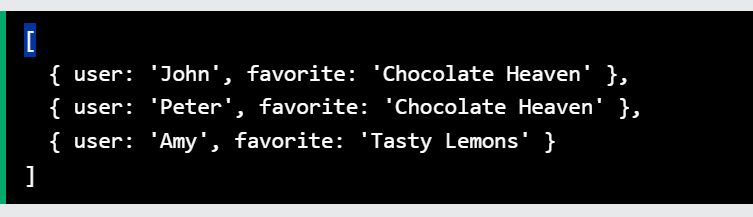
if (err) throw err;

console.log(result);

});

});

Which will give you this result:

[

**PART B**

**Aim: - Write a Program to work with NodeJS and Mysql**

1. Write a Program to create a Database “Employee\_DB” using Nodejs Script.

SHOW DATABASES;

create database Employee\_DB;

1. Write a Program to create a Table “Employee” using Nodejs Script.

**Table Columns**

**Employee\_Id,Employee\_Name,Employee\_Address, Employee\_Position and Employee\_Salary**

var mysql      = require('mysql');

var con = mysql.createConnection({

  host     : 'localhost',

  user     : 'root',

  password : 'root',

  port: 3308,

  database : 'Employee\_DB'

});

con.connect();

console.log("Connected!");

var sql = "CREATE TABLE Employee (Employee\_ID int(4), Employee\_Name VARCHAR(255), Employee\_Address VARCHAR(255), Employee\_Postion VARCHAR(255), Employee\_Salary  float(10,2))";

    con.query(sql, function (*err*, *result*) {

      if (*err*) throw *err*;

      console.log(*result*);

    });

use Employee\_DB;

select \* from Employee;

1. Write a Program to insert 7 Employee Details into a Table “Employee” using Nodej Script and Each Employee\_Id should not be Null and should be Unique.

var mysql      = require('mysql');

var con = mysql.createConnection({

  host     : 'localhost',

  user     : 'root',

  password : 'root',

  port: 3308,

  database : 'Employee\_DB'

});

con.connect();

console.log("Connected!");

var sql = "INSERT INTO Employee VALUES (4001, 'PRADEEP', 'A', 'H.R', 86000.00), (4002, 'ASHOK', 'B', 'MANAGER', 52028.00), (4003, 'PAVAN KUMAR', 'C', 'ASST MANAGER', 28000.00), (4004, 'SANTHOSH', 'D', 'STORE MANAGER', 9500.00), (4005, 'THAMAN', 'E', 'GENERAL MANAGER', 26000.00), (4006, 'RAMAN', 'F', 'GENERAL MANAGER', 26000.00), (4007, 'THOMAS', 'G', 'VP', 260000.00)";

    con.query(sql, function (*err*, *result*) {

      if (*err*) throw *err*;

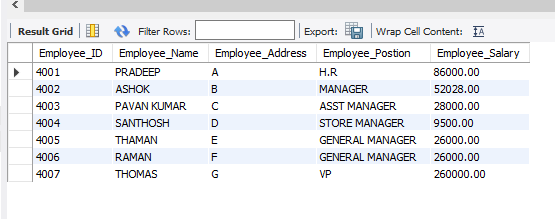
      console.log(*result*);

    });

1. Write a Program to display all Employee Details from Table Employee.

use Employee\_DB;

select \* from Employee;



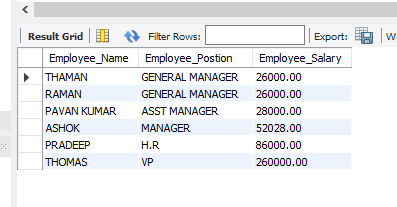
1. Write a Program to Display Employee\_Name, Employee\_Position and Employee\_Salary whos Salary is more than 10000 in ascending order.

SELECT Employee\_Name, Employee\_Postion, Employee\_Salary

FROM Employee

WHERE Employee\_Salary > 10000

ORDER BY Employee\_Salary ASC;



1. Create a one more Table call Employee\_Profile where columns are Employee\_Id, Employee\_Profile and Employee\_Name.

var mysql      = require('mysql');

var con = mysql.createConnection({

  host     : 'localhost',

  user     : 'root',

  password : 'root',

  port: 3308,

  database : 'Employee\_DB'

});

con.connect();

console.log("Connected!");

var sql = "CREATE TABLE Employee\_Profile( Employee\_Id int(4) AUTO\_INCREMENT PRIMARY KEY, Employee\_Profile VARCHAR(255) NOT NULL, Employee\_Name VARCHAR(50) NOT NULL )";

    con.query(sql, function (*err*, *result*) {

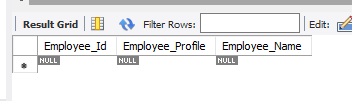
      if (*err*) throw *err*;

      console.log(*result*);

    });

use Employee\_DB;

select \* from Employee\_Profile;



1. Write a program to join these two table and display Employee\_Name , Employee\_Position ,Employee\_Profile and Employee\_Salary based on Employee\_Id.

**SELECT**

**ep.Employee\_Name,**

**e.Employee\_Postion,**

**ep.Employee\_Profile,**

**e.Employee\_Salary**

**FROM**

**Employee\_Profile ep**

**JOIN**

**Employee e ON ep.Employee\_Id = e.Employee\_Id;**

var mysql = require('mysql');

var con = mysql.createConnection({

    host: 'localhost', user: 'root', password: 'root', port: 3306,

    database: 'Employee\_DB'

});

con.connect(function(*err*) { if (*err*) throw *err*; });

console.log("Connected!");

var sql = "SELECT e.Employee\_Id, e.Employee\_Name, p.Employee\_Profile FROM EMPLOYEE e LEFT JOIN Employee\_Profile p ON e.Employee\_Id = p.Employee\_Id;";

con.query(sql, function (*err*, *result*) { if (*err*) throw *err*; console.log(*result*); });

**Observation and Learning: -**

* **Write your observation and learning**

**Answer the Questions: -**

**Conclusion: Write your learning about Nodejs and Mysql Database Connection.**