7...

Database Connectivity

Objectives...

- To establishing connection with database.
- To creating database through MySQL dashboard and query.
- To creating Schema / Tables.
- To insert records and display records from table.
- To update and delete records from table.

7.1 INTRODUCTION

- A Database connection is a facility in computer science that allows client software to talk to database server software, whether on the same machine or not. A connection is required to send commands and receive answers, usually in the form of a result set.

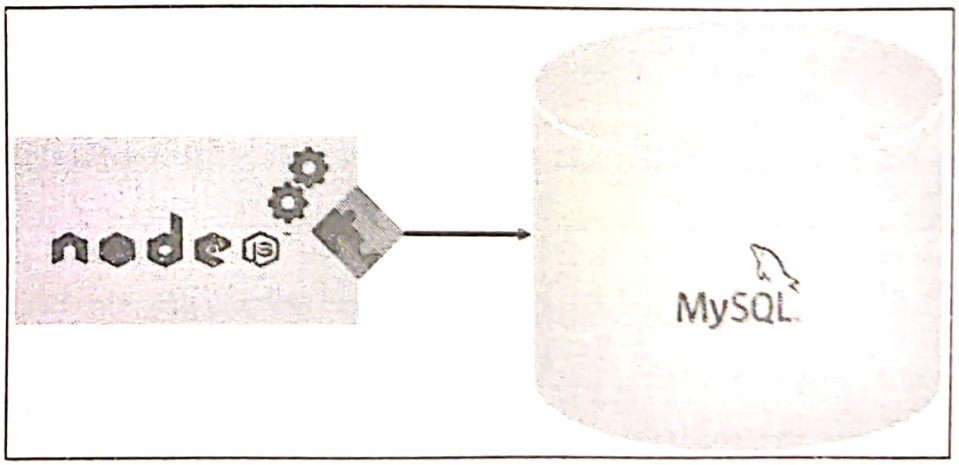


Fig. 7.1: Front End and Back End

- Connections are built by supplying an underlying driver or provider with a connection string, which is a way of addressing a specific database or server.

- Database connectivity with MySQL server using NODE JS is quite simple. We simply need to connect to the MySQL database server from a Node. Make sure that your system is having MySQL installed. If it is not, then do download the appropriate version from the link given below:

`https://www.mysql.com/downloads/`

OR

`https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-community-8.0.19.0.msi`

- If your system is installed with WampServer or Xampp Sever, then it is also okay for us to have database connectivity with MySQL server. Because both these servers are having MySQL built-in available in their set up as one of the services.

7.2 CONNECTION STRING [S-22, S-23]

Follow the steps given below for Database Connectivity:

Step 1: Create a project folder with valid and meaningful name under any available drives. For example: We create one folder with name **BCA** under **C drive.**

Step 2: Install MySQL Driver. To access a MySQL database with Node JS, we need a MySQL driver.

We will be using the "mysql" module, downloaded from NPM(Node Package Manager).

Now, go to command prompt and execute the following command:

```
C:\Users\UserName>npm install mysql
```

When you execute above command, you may see following details. It shows that you have successfully installed MySQL database driver.

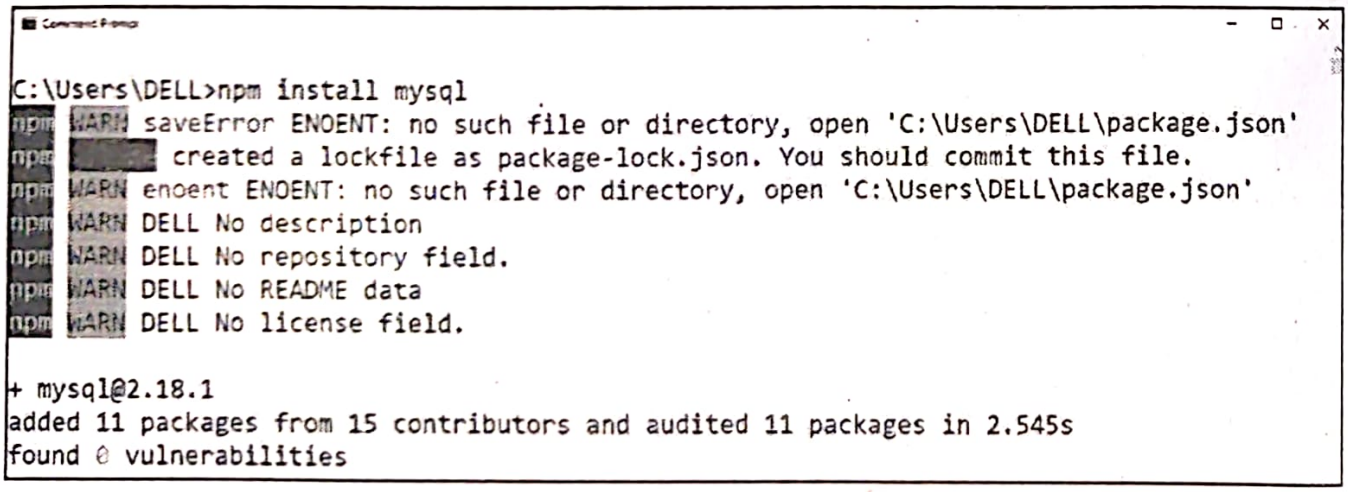


Fig. 7.2 : Installation of MySQL driver for Node JS

Step 3: Creating a Connection: Now, we should write a code to connect with the MySQL database server. Write following code in a notepad or any editor and save it under **BCA** folder with the name **db_connect.js**

Program 7.1: Program for creating database connection.

db_connect.js

```
// include mysql module
var mysql = require(mysql);

// create a connection variable with the required details
var con = mysql createConnection({
    host: "localhost",          // IP address of server running mysql
    user: "root",               // user name to your mysql database
    password: "iccs#123",       // corresponding password.
    port : '3308'               /* Optional. But mention it if multiple
                                servers(such as MariaDB, MySQL) are running
                                simultaneously */
});

// make connection to the database
con.connect(function(err) {
    if (err) throw err;
    // if connection is successful
    console.log("Connected to the MySQL server!");
});

//Closing the connection
con.end(function(err) {
    if (err) {
          return console.log('error:' + err.message);
    }
    console.log('Closed the active database connection.'};
});
```

Output:

```
C:\BCA>node db_connect.js
Connected to the MySQL server!
```

Step 4: Now, as the connection is established successfully with MySQL server database, we can start querying the database using SQL statements.

7.3 CONFIGURING

7.3.1 Creating Database in MySQL Server

A. Method : 1(Using Dashboard of MySQL Server)

- To create a new database using the MySQL Workbench, follow these simple steps:

1. **First, launch the MySQL Workbench.**

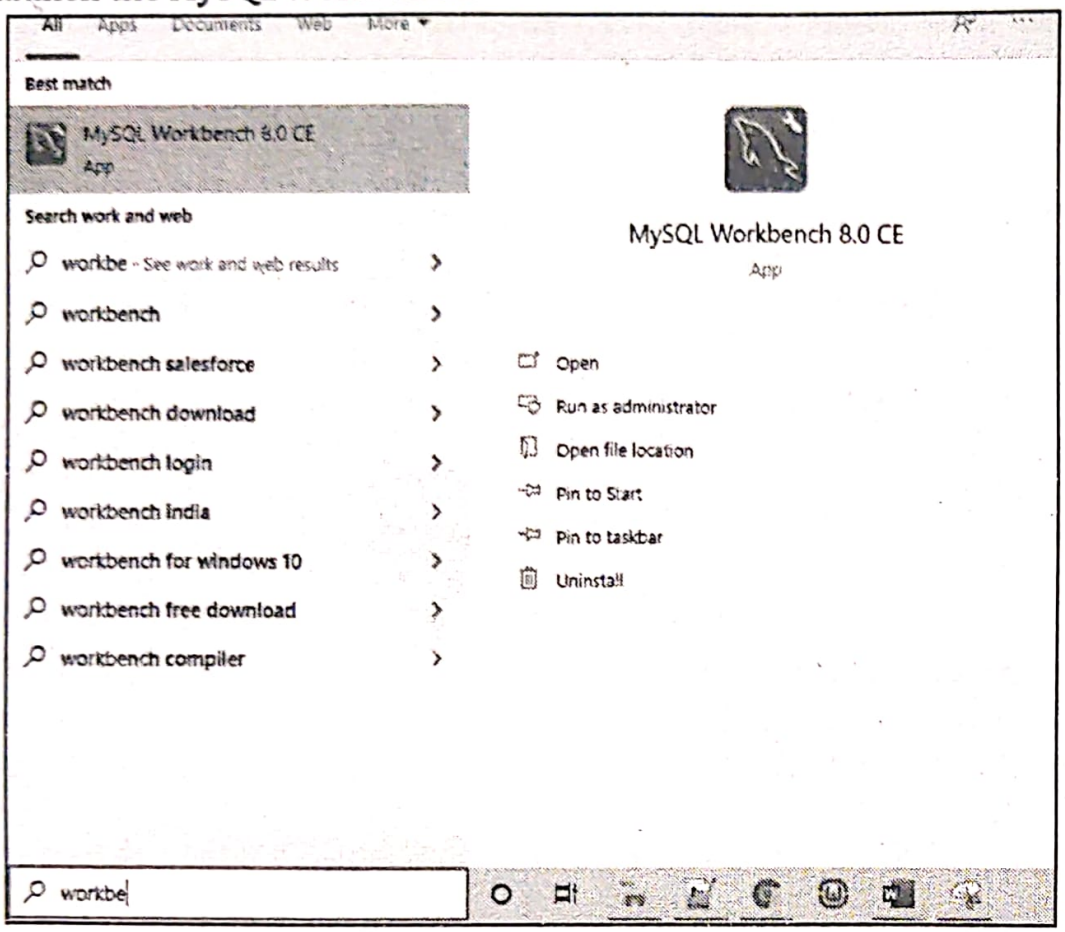


Fig. 7.3

2. **Click the setup new connection button as shown in the following screenshot.**

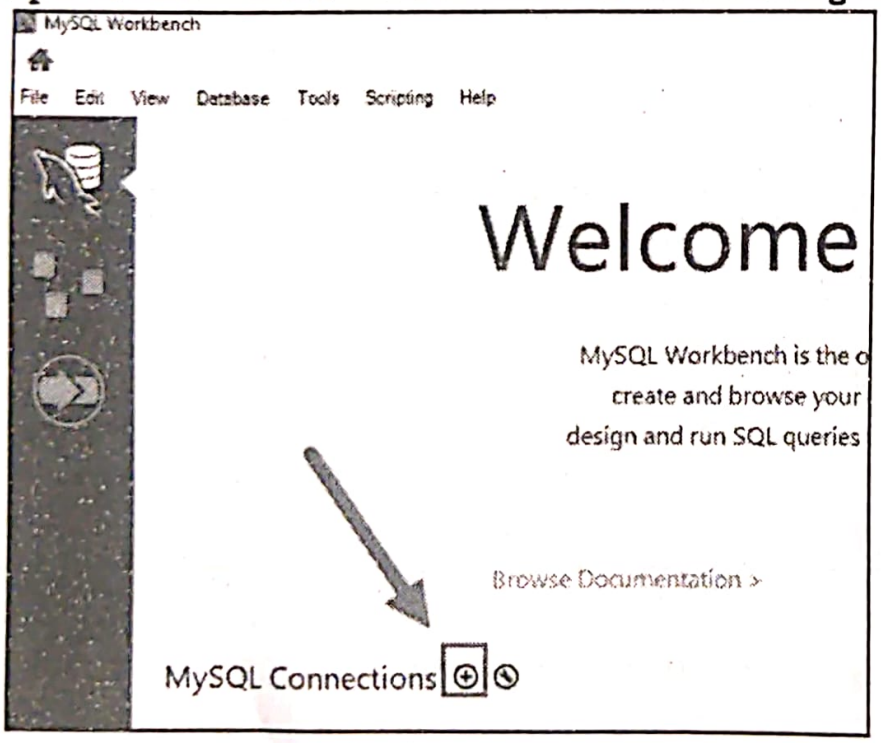


Fig. 7.4

3. Type the name for the connection and click the Test Connection button.

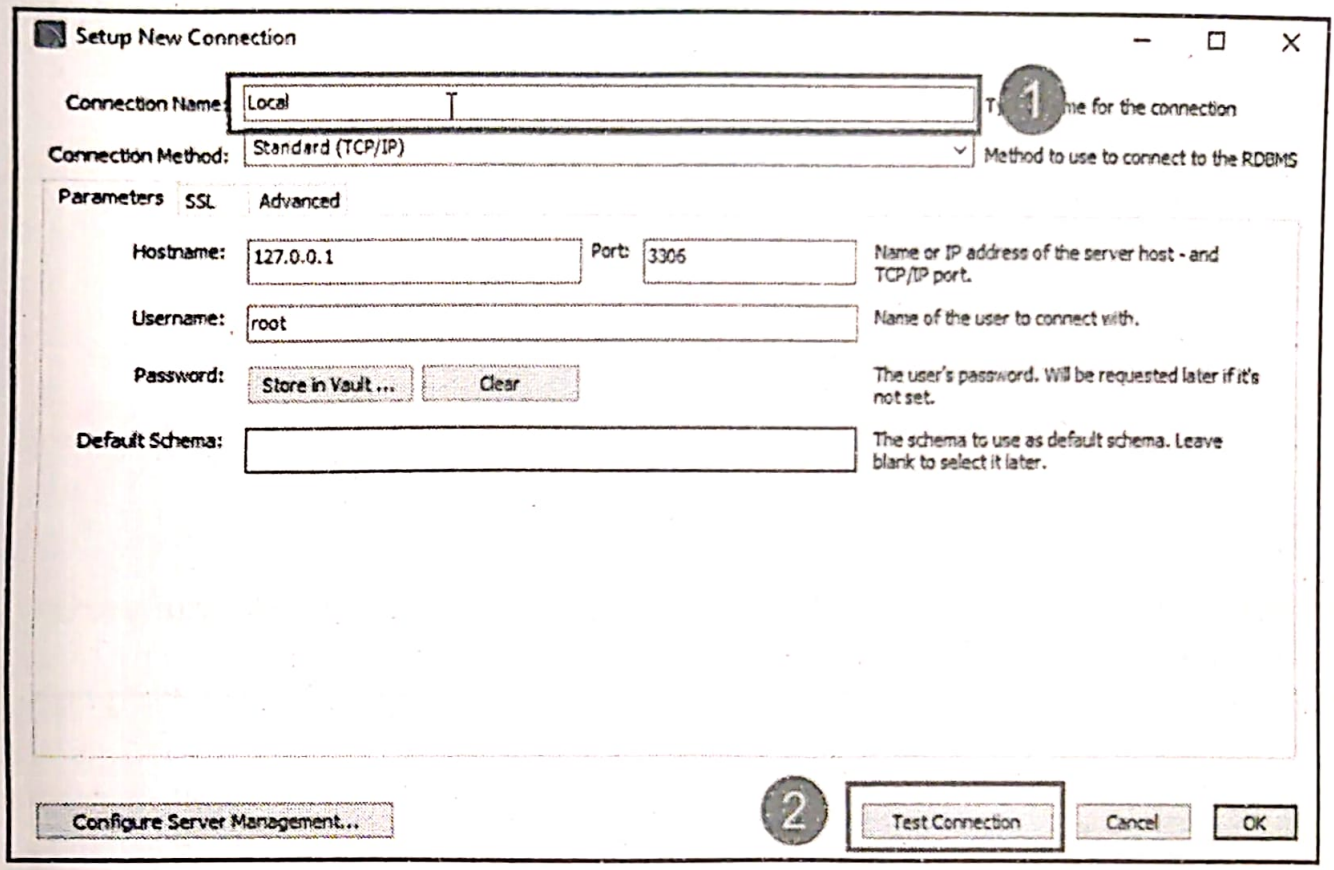


Fig. 7.5

4. MySQL Workbench displays a dialog asking for the password of the root user.

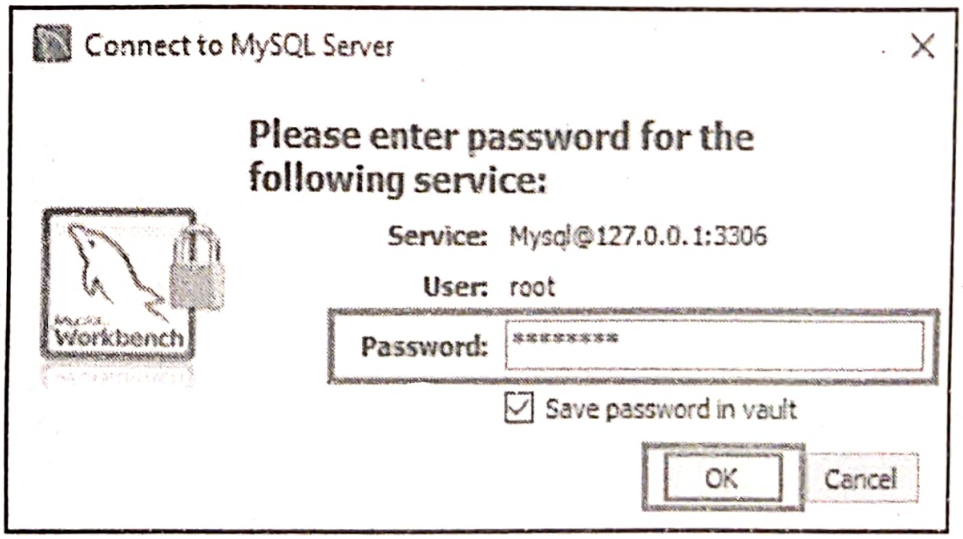


Fig. 7.6

You need to type the password for the root user, then check the Save password in vault, and click OK button.

5. **Double click the connection name Local to connect to the MySQL Server.**

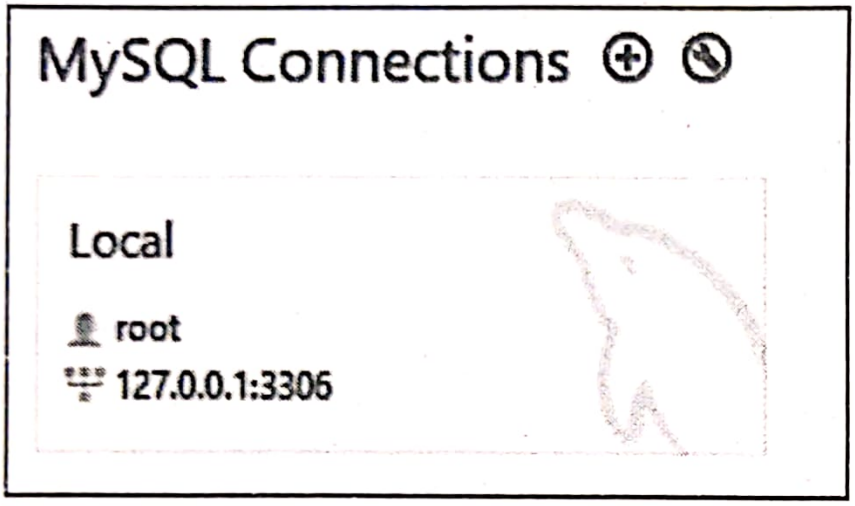


Fig. 7.7

6. **MySQL Workbench opens the following window which consists of four parts: Navigator, Query, Information, and Output.**

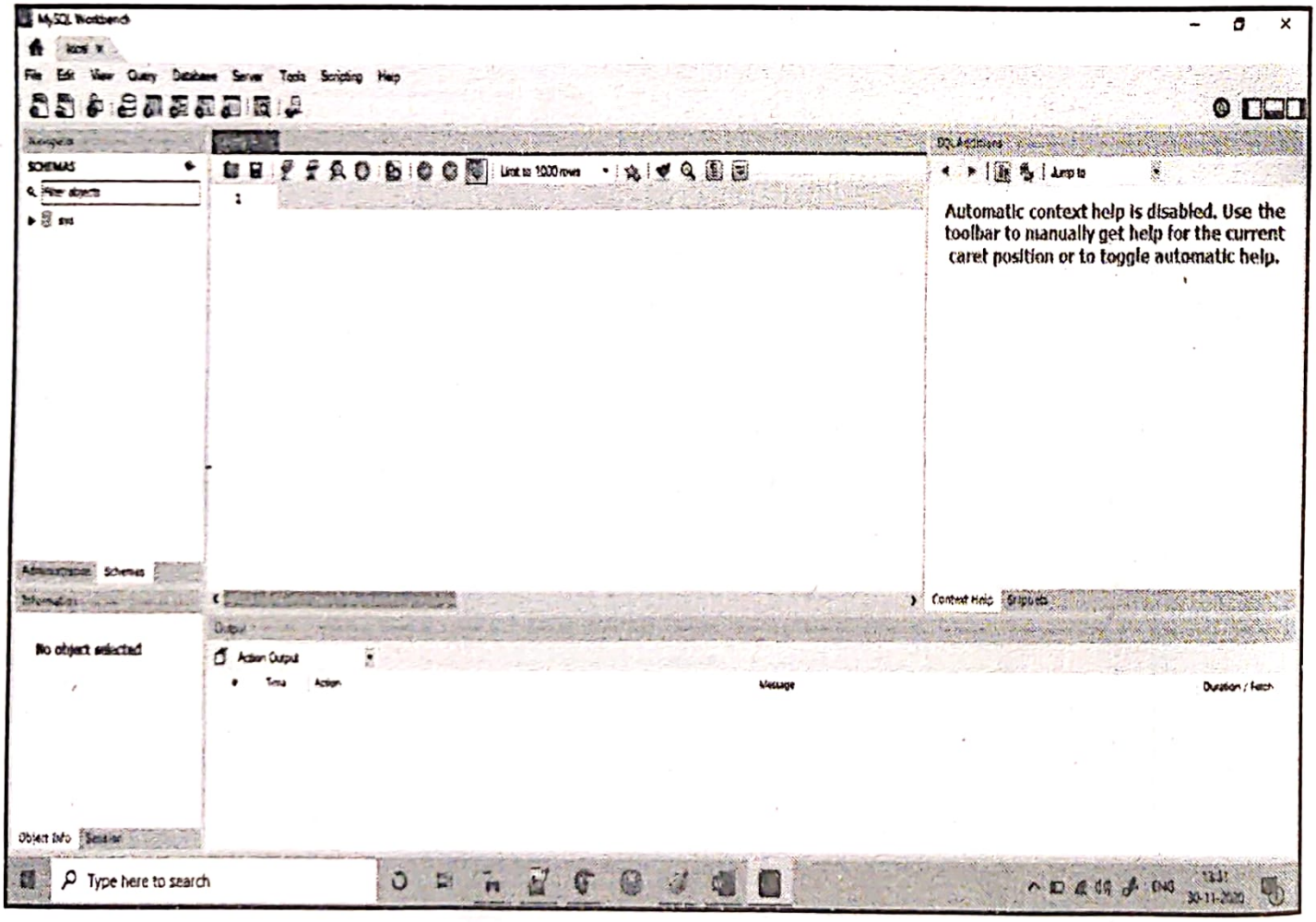


Fig. 7.8

7. Click the create a new schema in the connected server button from the toolbar.

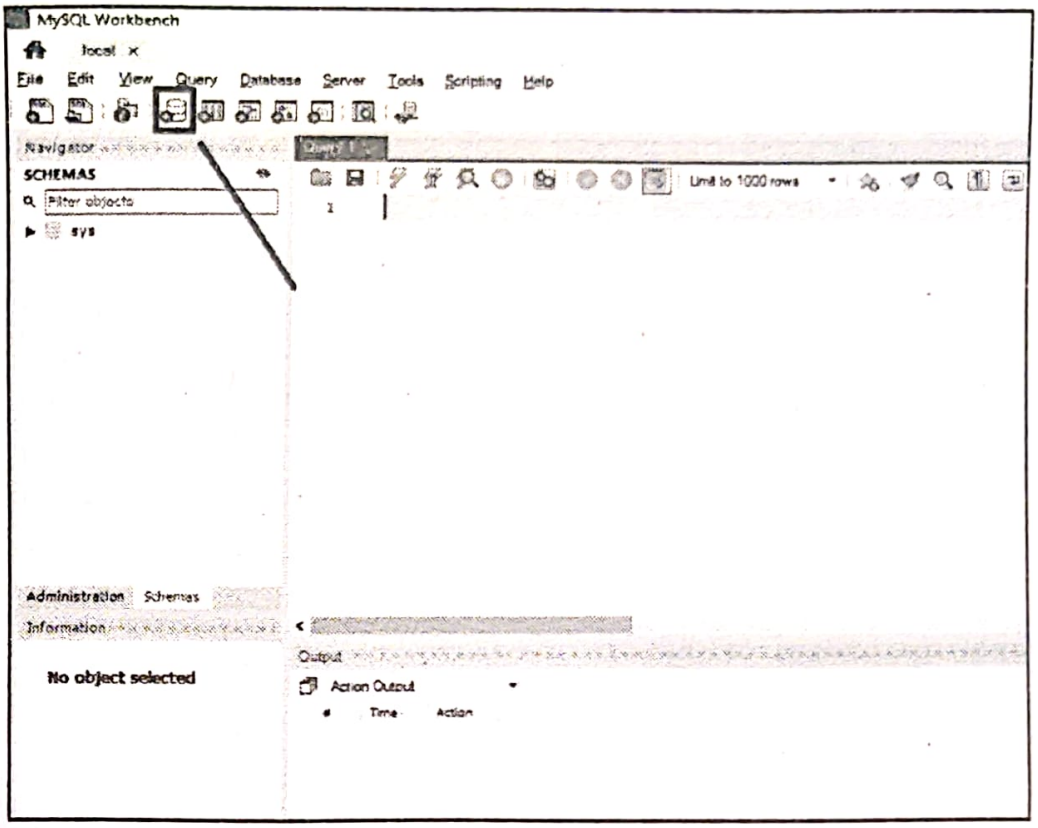


Fig. 7.9

- In MySQL, the schema is the synonym for the database. Creating a new schema also means creating a new database.

8. The following window is open. You need to enter the schema name, then change the character set and collation if necessary, and click the Apply button.

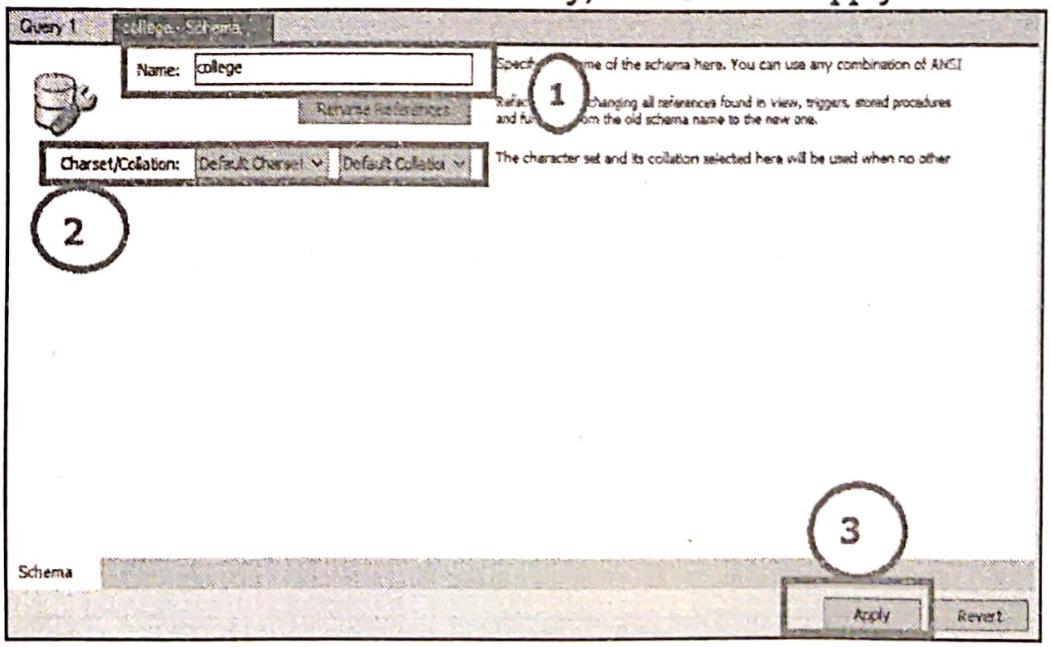


Fig. 7.10

9. MySQL Workbench opens the following window that displays the SQL script which will be executed. Note that the CREATE SCHEMA statement command has the same effect as the CREATE DATABASE statement.

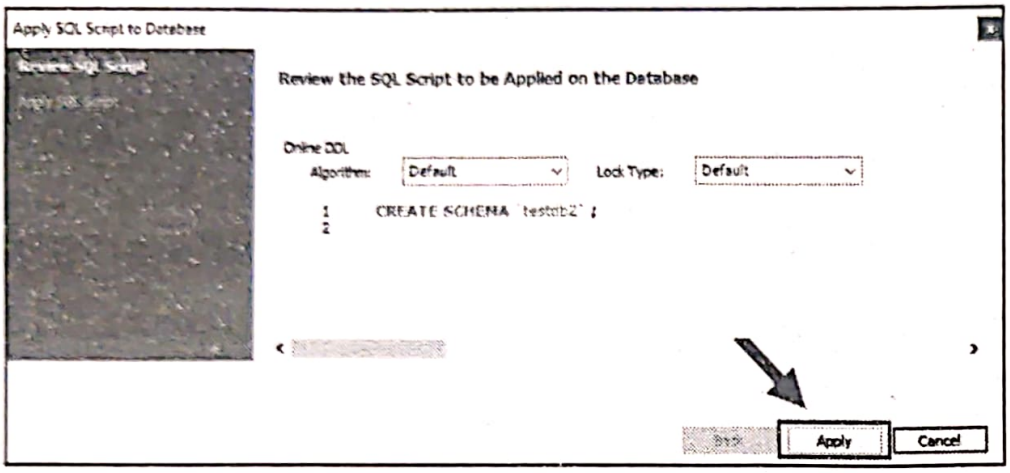


Fig. 7.11

10. Click on the Finish button.

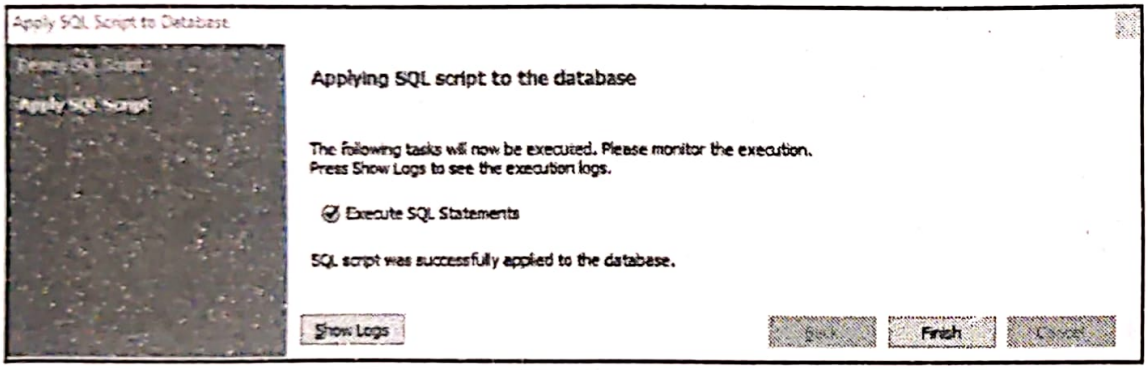


Fig. 7.12

11. If everything is fine, you will see the new database created and showed in the schemas tab of the Navigator section.

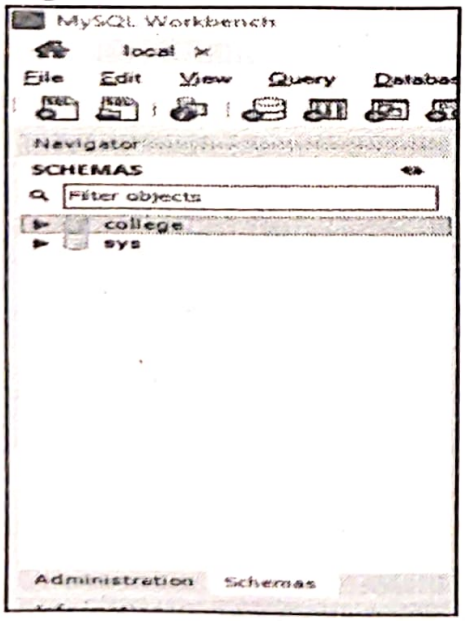


Fig. 7.13

12. To select the college database then right click the database name and choose Set as Default Schema menu item.

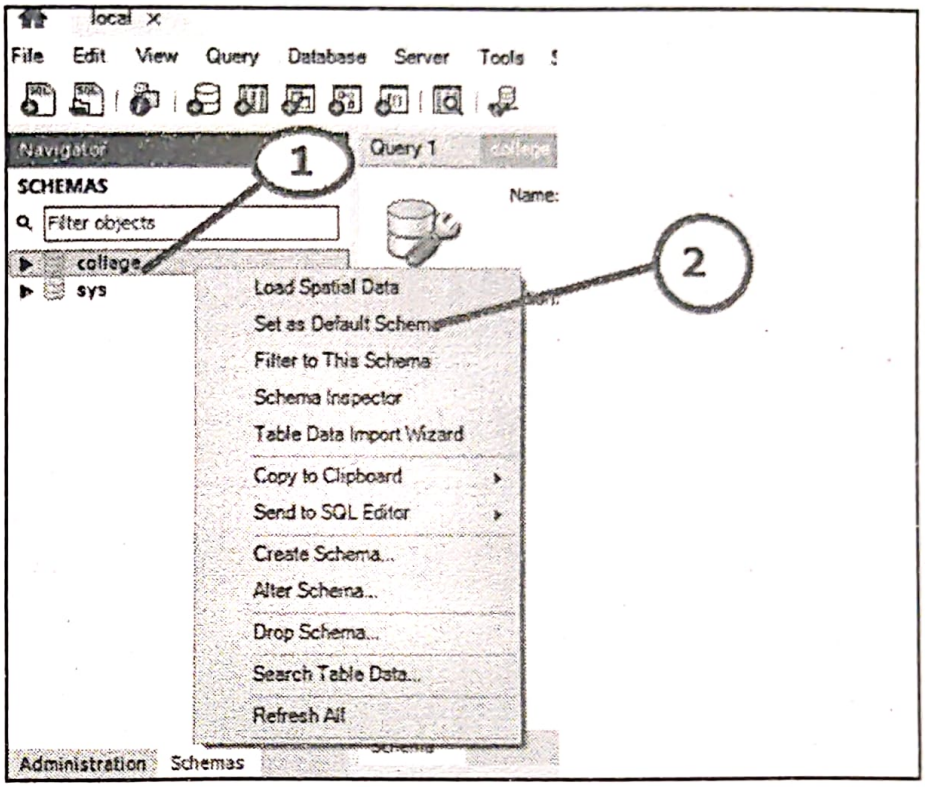


Fig. 7.14

13. The college node is open as shown in the following figure.

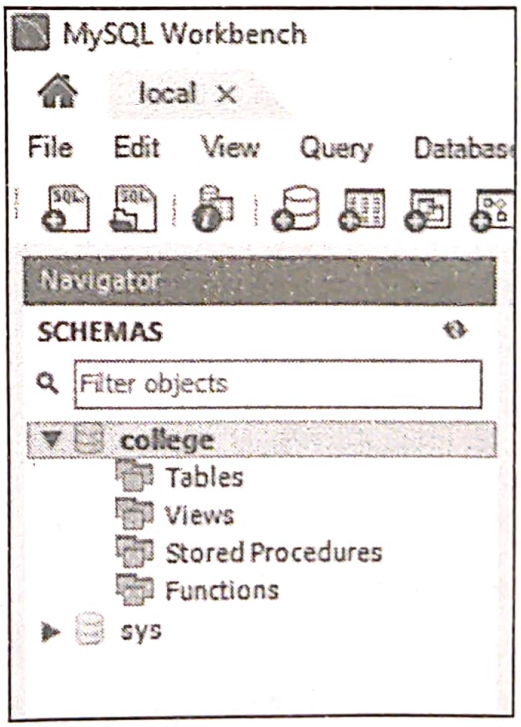


Fig. 7.15

14. Now, we can work with college from the MySQL Workbench.

B. Method : 2(Through Coding)

- Write following code in any editor, save it with the name **create_db.js** and execute this file from command prompt.

create_db.js

```
// include mysql module
var mysql = require('mysql');

var con = mysql.createConnection({
    host: "localhost",
    user: "root",
    password: "iccs#123",
    port : '3308'
});

//Making connection to database
con.connect(function(err) {
    if (err) throw err;
    console.log("Connected to the Database!");
    con.query("CREATE DATABASE college", function (err, result) {
        if (err) throw err;
        console.log("Database with name college created succssfully...");
    });
});
```

7.3.2 Creating a Table in the Database College

A. Method : 1(Using Dashboard of MySQL Server)

- To create new table under college database, follow simple steps given below.

1. Open the MySQL Server Workbench. And right click on the database name then click on Create table option.

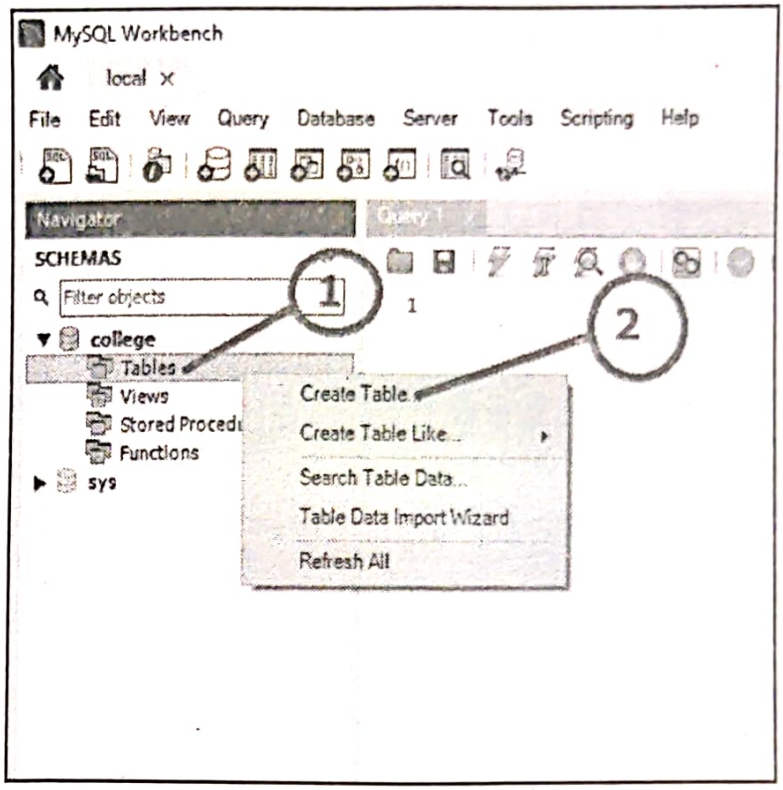


Fig. 7.16

2. **Now, 1st insert proper and meaningful name to the table. 2nd Create necessary fields with appropriate names and also select their type (datatype) then 3rd Click on Apply.**

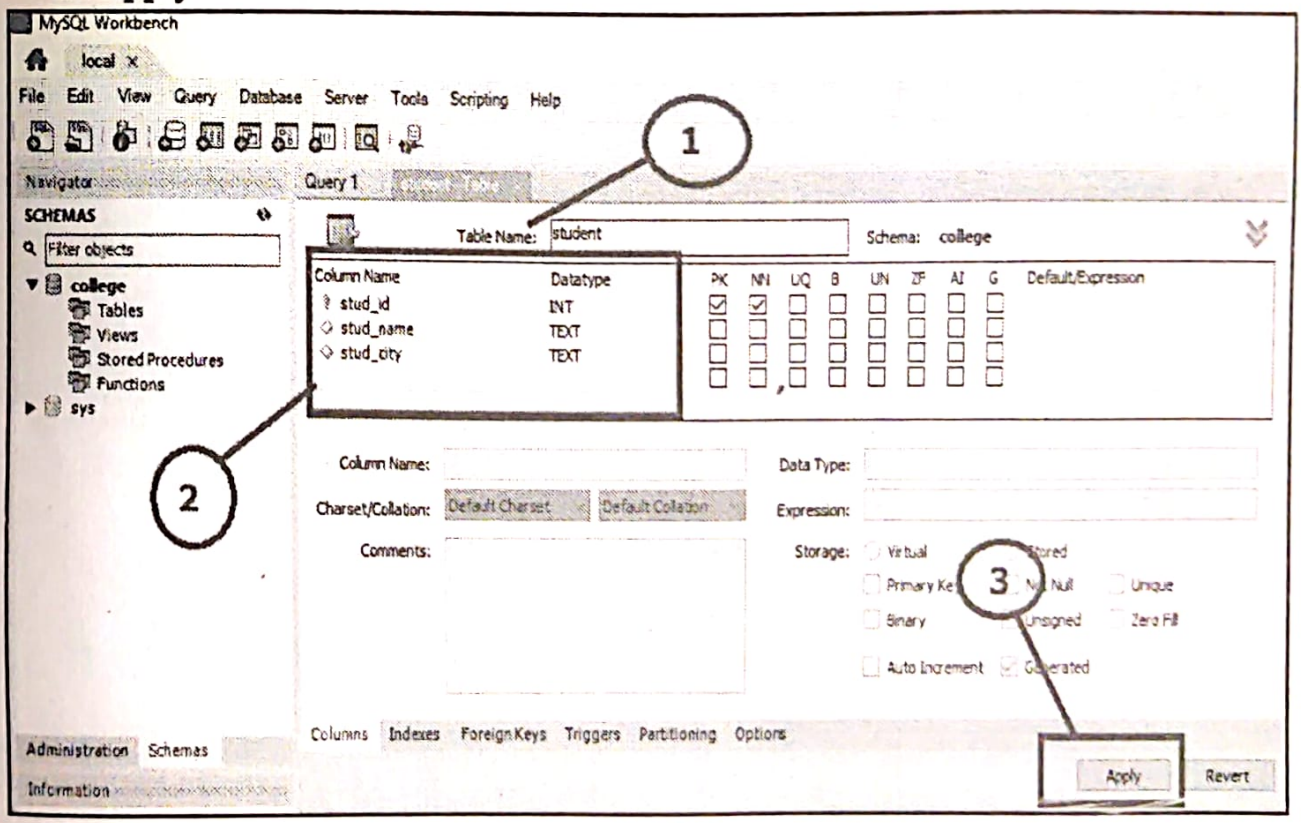


Fig. 7.17

3. Click on Apply button.

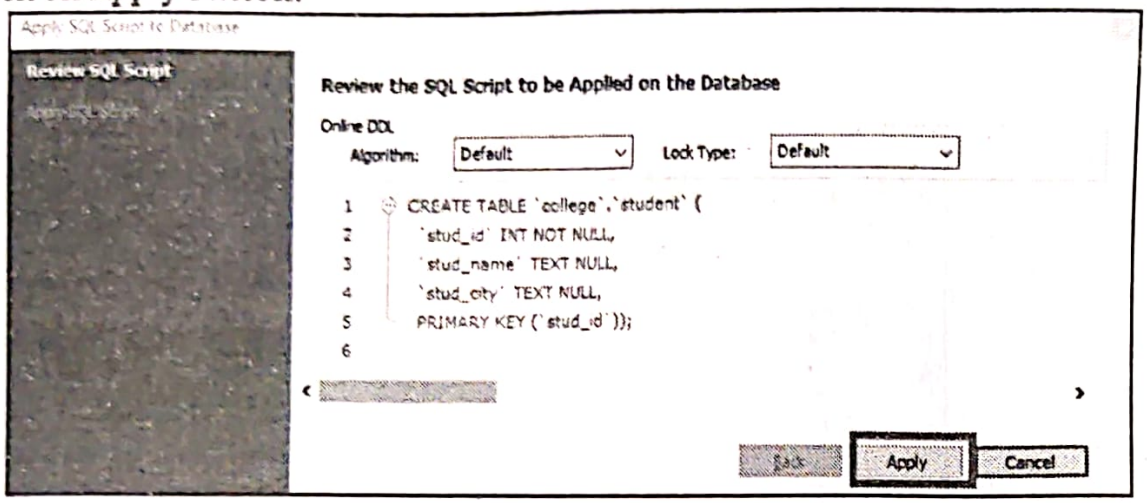


Fig. 7.18

4. Click on Finish button

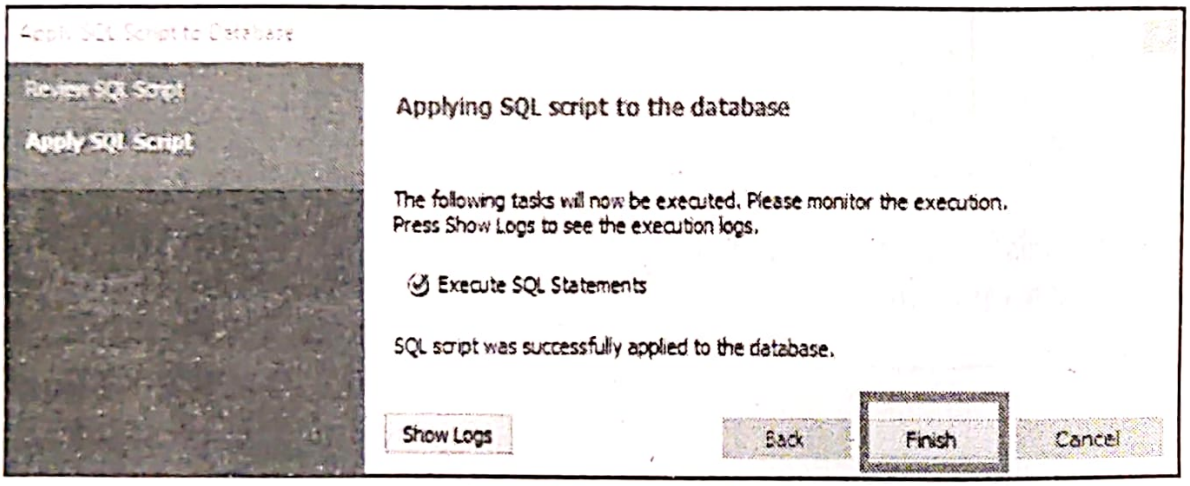


Fig. 7.19

5. Now, you can see table student got created under the database college.

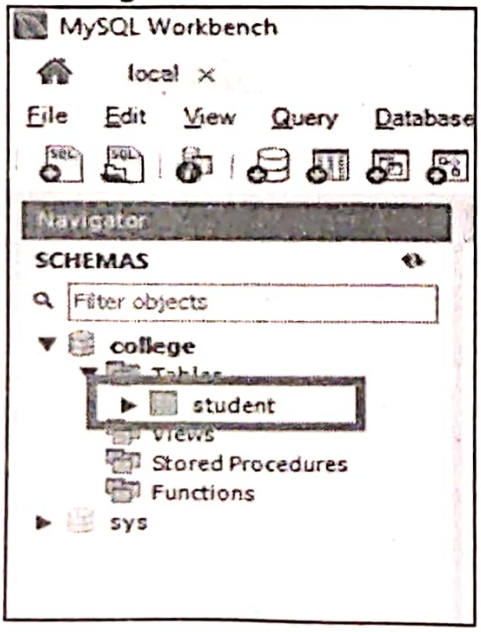


Fig. 7.20

B. Method : 2 (Through Coding)

- Here, we will be using "CREATE TABLE" statement of MySQL. Now, we need to specify the database name in which we are going to create a table.

Program 7.2: Program to create table student.

create_table.js

```
var mysql = require('mysg1');

var con = mysql.createConnection({
   host: "localhost",
   user: "root",
   password: "iccs#123",
   database: "college",
   port : '3308'
});

con.connect(function(err) {
   if (err) throw err;
   console.log("Connected to the Database!");
   var sql = "CREATE TABLE student(stud_roll int, stud_name text, stud_city
   text, PRIMARY KEY(stud_roll))";

   con.query(sql, function (err, result) {
      if (err) throw err;
      console.log("Table with name student created successfully...");
   });
});
```

Output:

Command Prompt - node create_table.js

```
C:\BCA>node create_table.js
Connected to the Database!
Table with name student created successfully...
```

- You can see the View/Structure of the student table in the MySQL dashboard as shown below:

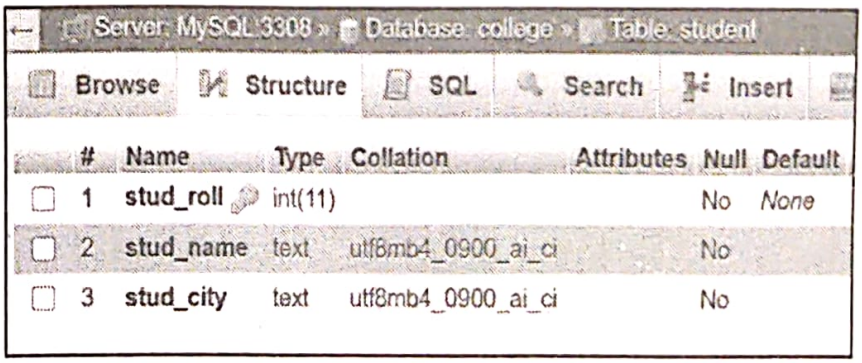


Fig. 7.21: student table

7.3.3 Insert a Record in the Table

Insert Query:

- The figure given below is the schema of table named **student** available under the database **college.** This table contains **3** columns namely **stud_roll, stud_name** and **stud_city.**

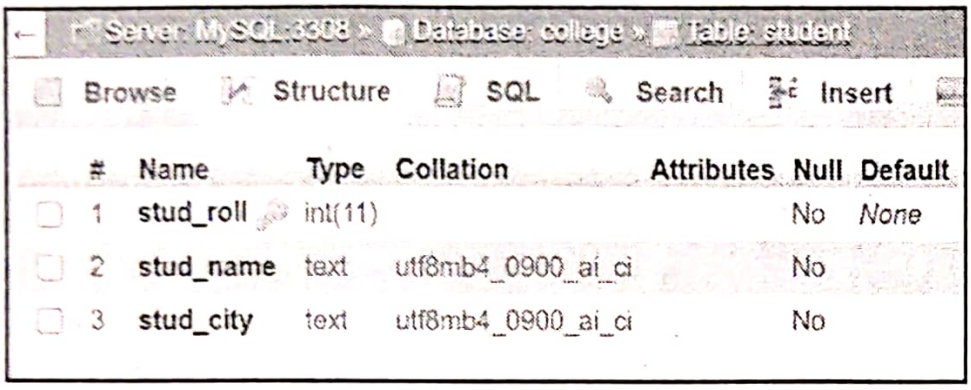


Fig. 7.22: student table

- Observe that, the field named **stud_roll** is having type **int** and is a primary key of this table student. Other two fields namely **stud_name** and **stud_city** are of the type **text.** None of the field is null. All fields are made compulsory.
- Now, we are going to write **insert** query against this table **student.** We need to provide 3 values to the query.

Program 7.3: Program to insert record in table student.

insert_query.js

```
//include Prompt-sync module
const prompt = require ('prompt-sync') ();

// include mysql module
var mysql = require('mysql');

// create a connection variable with the required details
var con = mysql.createConnection({
   host: "localhost",
   user: "root",
   password: "Iccs#123',
   database: "college",
   port : 3308
});

// make connection to the database
con.connect (function(err) {
   if (err) throw err;
```

```
    // if connection is successful
    console.log(Connected!');

    //Accepting the user Input from the user
    const roll = prompt('Enter Roll Number :');
    const name = prompt('Enter Student Name :');
    const city = prompt('Enter City :');

    //Designing Dynamic Query based on the inputs received from the user
    var sql = "INSERT INTO student (stud_roll, stud_name, stud_city) VALUES
    ("+ Number(roll) + ", '" + name + "', '" + city + "')";

    con.query(sql, function (err, result) {
        if (err) throw err;
        // if there is no error, you have the result
        console.log("1 record successfully inserted in the table");
    });
});
```

Output:

```
C:\Windows\System32\cmd.exe - node insert_query.js

C:\BCA>node insert_query.js
Connected!
Enter Roll Number :101
Enter Student Name :Janardan
Enter City :Pune
1 record successfully inserted in the table
```

- Assume that we have entered few more records by executing the same code. Image below shows the records currently available in the table **student**.

Table 7.1 : student

stud_roll	stud_name	stud_city
101	Janardan	Pune
102	Shivendu	Mumbai
103	Tejashri	Thane
104	Vishal	Varanasi
105	Shubhangi	Pune
106	Ashish	Nagpur

7.4 WORKING WITH SELECT COMMAND

- Now, we have some records in the table student. Let us write SELECT statement against this table to display all the available records.

Program 7.4: Program to display all the records available in the table student. **[W-22, S-23]**

show_records.js

```
var mysql = revire('mysql');
var con = mysql.createConnection({
   host: "localhost",
   user: "root",
   password: "iccs#123",
   database: "college",
   port : '3308'
});
con.connect(function(err) {
   if (err) throw err;
   con.query("SELECT * FROM student", function (err, result, fields) {
      if (err) throw err;
      //Showing all the records
      console.log(result);
   });
});
```

Output:

```
Command Prompt - node show_records.js
C:\BCA>node show_records.js
[
  RowDataPacket {
    stud_roll: 101,
    stud_name: 'Janardan',
    stud_city: 'Pune'
  },
  RowDataPacket {
    stud_roll: 102,
    stud_name: 'Shivendu',
    stud_city: 'Mumbai'
  },
  RowDataPacket {
    stud_roll: 103,
    stud_name: 'Tejashri',
    stud_city: 'Thane'
  },
  RowDataPacket {
    stud_roll: 104,
    stud_name: 'Vishal',
    stud_city: 'Varanasi'
  },
  RowDataPacket {
    stud_roll: 105,
    stud_name: 'Shubhangi',
    stud_city: 'Pune'
  },
  RowDataPacket {
    stud_roll: 106,
    stud_name: 'Ashish',
    stud_city: 'Nagpur'
  }
```

The Result Object:

- As you can see from the output of the example above, the **result** object is an array containing each row as an object.
- To return e.g. the **name** of the third record, just refer to the third array object's as :

```
console.log("Name is", result[2].stud_name);
```

- This will show the output as **Name is Tejashri**
- We can slightly modify above code and print all the records in proper format.

Program 7.5: Program to display all the records available in the table student in table format.

show_records_format.js

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

var con = mysql.createConnection({
   host: "localhost",
   user: "root",
   password: "iccs#123",
   database: "college",
   port : '3308'
});

con.connect(function(err) {
   if (err) throw err;
   con.query("SELECT * FROM student", function (err, result, fields) {
         if (err) throw err;
         //Showing records

         console.log("Roll No | Name | City");
         console.log("----------------------");

         //Code to iterate through an array
         Object.keys(result).forEach(function(key) {
            var record = result[key];  //Storing each row and accessing it
            console.log(record.stud_roll,"|", record.stud_name, "|",
            record.stud_city);
         });
   });
});
```

Output:

Command Prompt - node show_records_format.js

```
C:\BCA>node show_records_format.js
Roll No   |  Name |  City
-------------------------
101 | Janardan | Pune
102 | Shivendu | Mumbai
103 | Tejashri | Thane
104 | Vishal | Varanasi
105 | Shubhangi | Pune
106 | Ashish | Nagpur
```

7.5 UPDATING RECORDS

- Now, we have total SIX records in the table. Let us update the stud_city column. There are two records with the city name Pune. We will update value of city name from **Pune** to **Punya Nagari** using SQL update command. So, total TWO records will get updated.

Program 7.6: Program to update the student table records. **[W-22, S-23]**

update_table_query.js

```
const prompt = require('prompt-sync')();
var mysql = require('mysql');
var con = mysql.createConnection({
   host: "localhost",
   user: "root",
   password: "iccs#123",
   database: "college",
   port : '3308'
});

con.connect(function(err) {
   if (err) throw err;
   const old_city = prompt("Enter Current City Name to be changed ::");
   const new_city = prompt("Enter New City Name ::");

   var sql = "UPDATE student SET stud_city='" + new_city + "' WHERE
   stud_city='" + old_city + "'";

   //Executing the above query
   con.query(sql, function (err, result) {
      if (err) throw err;
      console.log(result.affectedRows + " record(s) updated");
   });
});
```

Output:

```
C:\Windows\System32\cmd.exe - node update_table_query.js

C:\BCA>node update_table_query.js
Enter Current City Name to be changed ::Pune
Enter New City Name ::Punya Nagari
2 record(s) updated
```

- **Result** object's affectedRows property returns the number of rows(records) affected due to this update query.

7.6 DELETING RECORDS

- Now, delete record(s) using SQL DELETE statement where city name is **Thane.**

Program 7.7: Program to delete the table record where city name is Thane. W-22

delete_record.js

```
const prompt = require('prompt-sync')();

var mysql = require('mysql');

var con = mysql.createConnection({
   host: "localhost",
   user: "root",
   password: "iccs#123",
   database: "college",
   port: '3308'
});

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

   const cityName = prompt("Enter City Name::");

   var sql = "DELETE from student WHERE stud_city ='" + cityName + "'";

   //Executing the above query
   con.query(sql, function (err, result) {
      if (err) throw err;
      console.log(result.affectedRows + " record(s) deleted");
   });
});
```

Output:

```
C:\Windows\System32\cmd.exe - node delete_record.js

C:\BCA>node delete_record.js
Enter City Name ::Thane
1 record(s) deleted
```

Solved Programs

Program 7.8: Program to print all the records of the students who are living in the city "Pune". Now, let us assume following table **student.**

stud_roll	stud_name	stud_city	stud_per
101	Janardan	Pune	78.56
102	Shivendu	Mumbai	69.85
103	Tejashri	Thane	72.62
104	Vishal	Varanasi	65.46
105	Shubhangi	Pune	71.29
106	Ashish	Nagpur	63.98

print_city_records.js

```
const prompt = require('prompt-sync')();
var mysql = require('mysql');

var con = mysql.createConnection({
   host: "localhost",
   user: "root",
   password: "iccs#123",
   database: "college",
   port : '3308'
});

con.connect(function(err) {
   if (err) throw err;
   const cityName = prompt("Enter City Name ::");

   con.query("SELECT * FROM student where stud_city ='" + cityName + "'",
   function (err, result, fields) {
      if (err) throw err;
      //Showing records

      console.log("Roll | Name | City | Percentage");
      console.log("------------------------------");

      //Code to iterate through an array
      Object.keys(result).forEach(function(key){
         var record = result[key]; //Storing each row and accessing it
         console.log(record.stud_roll,"|", record.stud_name, "|",
         record.stud_city, "|", record.stud_per);
      });
   });
});
```

Output:

C:\Windows\System32\cmd.exe - node print_city_records.js

```
C:\BCA>node print_city_records.js
Enter City Name ::Pune
Roll|     Name   | City | Percentage
--------------------------
101 | Janardan | Pune | 88.23
105 | Shubhangi| Pune | 88.69
```

Program 7.9: Program to print the merit list of the college. All students have earned marks and thus they got the percentage. Now, we will print all the records in the descending order of their percentage, which is nothing but our merit list.

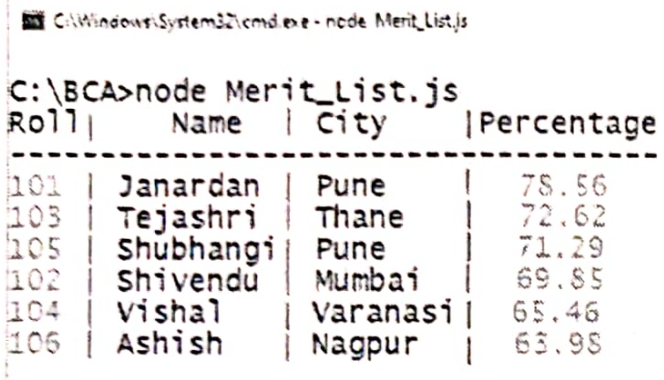
Merit_List.js

```
var mysql = require('mysql');
var con = mysql.createConnection({
   host: "localhost",
   user: "root",
   password: "iccs#123",
   database: "college",
   port : '3308'
});

con.connect(function(err) {
   if (err) throw err;
   con.query("SELECT * FROM student order by stud_per desc", function (err,
   result, fields) {
      if (err) throw err;
      //Showing records in desceding order of the percentage
      console.log("Roll | Name | City | Percentage");
      console.log("-----------------------------");

      //Code to iterate through an array
      Object.keys(result).forEach(function(key) {
            var record = result[key]; //Storing each row and accessing it
            console.log(record.stud_roll," ", record.stud_name, "|",
            record.stud_city, " ", record.stud_per);
      });
   });
});
```

Output:



Program 7.10: Program to print details of the first two toppers of the college using LIMIT statement. And also explain use of LIMIT statement.

LIMIT statement: We can limit the number of records returned from the query, by using the "LIMIT" statement:

toppers.js

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

var con = mysql.createConnection({
   host: "localhost",
   user: "root",
   password: "iccs#123",
   database: "college",
   port : '3308'
});

con.connect(function(err) {
   if (err) throw err;
   con.query("SELECT * FROM student order by stud_per DESC LIMIT 2",
   function (err, result, fields) {
      if (err) throw err;
      //Showing TOP TWO records of the meritorious students

      console.log("Roll | Name | City | Percentage");
      console.log("-----------------------------");

      //Code to iterate through an array
      Object.keys(result).forEach(function(key) {
         var record = result[key]; //Storing each row and accessing it
         console.log(record.stud_roll,"|", record.stud_name, "|",
         record.stud_city, "|", record.stud_per);
      });
```

```
    });
});
```

Output:

```
C:\Windows\System32\cmd.exe - node toppers.js

C:\BCA>node toppers.js
Roll|    Name   | City | Percentage
-------------------------------
101 | Janardan | Pune | 78.56
103 | Tejashri | Thane| 72.62
```

Program 7.11: Program to print Players details whose name ends with string "kar". Let us consider following table **Players.**

p_id	p_name	p_city
101	Sunil Gavaskar	Mumbai
102	Kapil Dev	Chandigarth
103	Dilip Vengsarkar	Pune
104	Sachin Tendulkar	Mumbai
105	Rahul Dravid	Bengaluru

player_records.js

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

var con = mysql.createConnection({
    host: "localhost",
    user: "root",
    password: "iccs#123",
    database: "college",
    port: "3308"
});

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

    con.query("SELECT * FROM players where p_name LIKE '%kar'", function
    (err, result, fields) {
        if (err) throw err;
        //Showing records of the players whose name ends with "kar"
```

```
        console.log("Player ID | Player Name | City");
        console.log("-----------------------------");

        //Code to iterate through an array
        Object.keys(result).forEach(function(key) {
           var record = result[key]; //Storing each row and accessing it
           console.log(record.p_id,"|", record.p_name, "|", record.p_city);
        });
     });
  });
```

Output:

```
C:\Windows\System32\cmd.exe - node player_records.js

C:\BCA>node player_records.js
Player ID|    Player Name   | City
-------------------------------------
101      | Sunil Gavaskar   | Mumbai
103      | Dilip Vengsarkar| Pune
104      | Sachin Tendulkar| Mumbai
```

Summary

- Mostly all modern day web applications have some sort of data storage system at the backend to store data. For example, if you take the case of a web shopping application, data such as the price of an item or the number of items of a particular type would be stored in the database.
- The Node JS framework has the ability to work with databases which are commonly required by most modern day web applications.
- Node JS can work with both relational (such as Oracle and MySQL Server) and non relational databases (such as MongoDB and MySQL).

Check Your Understanding

1. _____ method() of mysql module is used to establish a connection to the database in Node JS.
   (a) makeConnection() (b) createConnection()
   (c) sqlConnection() (d) connect()
2. Method used to terminate the connection with database server is _____.
   (a) close() (b) exit()
   (c) end() (d) terminate()
3. Apart from end(), an alternative way to end the connection is to call the _____ method.

(a) destroy() (b) exit()
(c) close() (d) terminate()

4. Which of the following is right command to install mysql database driver in Widows OS through command prompt?
(a) npm install mssql (b) npm install mysql
(c) npm install -mysql (d) None of the above

5. _____ property returns the number of rows altered/updated/affected due to INSERT, DELETE or UPDATE command.
(a) affectedRows() (b) AffectedRows
(c) AffectedRows() (d) affectedRows

6. To execute commands against the database table, we make use of _____ method.
(a) executeQuery() (b) query()
(c) sqlQuery() (d) Query()

Consider this database table "Players" and answer the questions given below:

p_id	p_name	p_city	runs	wickets
101	Sunil Gavaskar	Mumbai	5682	0
102	Kapil Dev	Chandigarth	3057	434
103	Dilip Vengsarkar	Pune	8630	0
104	Sachin Tendulkar	Mumbai	15786	153
105	Rahul Dravid	Bengaluru	12695	3

7. How many records will be retrieved after executing following SQL query?
SELECT * FROM players WHERE p_name LIKE 'S%ar'
(a) 2 (b) 1
(c) 5 (d) None

8. What will be the output of following SQL query?
SELECT COUNT(DISTINCT p_city) FROM players
(a) 2 (b) 1
(c) 4 (d) 5

9. Which of the following is the best and suitable mysql query to return details of leading wicket taker?
(a) SELECT * FROM `players` ORDER BY wickets DESC LIMIT 1
(b) SELECT * FROM `players` ORDER BY wickets DESC TOP 1
(c) SELECT * FROM `players` ORDER BY wickets ASC LIMIT 1
(d) SELECT * FROM `players` where wickets = MAX(wickets)

10. How many records will be fetched by following SQL query?

```
SELECT * FROM players WHERE p_city IN ('Pune', 'Mumbai')
```

(a) 2 (b) 0

(c) 4 (d) 3

ANSWER KEY

1. (b)	2. (c)	3. (a)	4. (b)	5. (d)
6. (b)	7. (a)	8. (c)	9. (a)	10. (a)

Practice Questions

Q.I: Answer the following questions in short.

1. Explain the working of createConnection() ?
2. Explain the parameters of createConnection() ?
3. Write a sql query to fetch all details of student from student table(details: stud_roll, Stud_name, Stud_addr, stud_) ?
4. What is connection string?

Q.II: Answer the following questions.

1. Write steps to Connect with database through code?
2. Write steps to connect to database using Dashboard?
3. Write Node JS Database connectivity Program to insert record ?
4. What is Database? Explain different functions in NODE JS to connect to the MySQL database server.
5. Write a program using Node JS to create a table Players set a constraint NULL. Columns/fields should include Player_ID, Player_Name, Sports_Name, and Country_Name.
6. Write a SQL statement to create a table named jobs including columns job_id, job_title, min_salary, max_salary and check whether the max_salary amount exceeding the upper limit Rs. 15000/-
7. Write a program using Node JS to print all the distinct cities from the table customers (cust_id, cust_name, cust_mail, cust_city).
8. Write a program using Node JS to update value of city from Allahabad to Prayagraj of all the customers (cust_id, cust_name, cust_mail, cust_city).
9. Write a program using Node JS to print records of those account holders whose balance amount is between Rs. 15,000 and Rs. 25,000/-. Assume suitable table schema.

❖❖❖

SOLVED UNIVERSITY QUESTION PAPERS

SUMMER 2022

Time : 2½ Hours] **[Max. Marks : 70**

Instructions to the candidates:

1) **All questions are compulsory.**
2) **Figures to the right indicate full marks.**

---

1. Answer the following (any Eight): **[8 × 2 = 16]**

a) What is Node. JS?

Ans: Refer to Section 1.2

b) Which type of applications we build using Node. JS?

Ans: Refer to Section 1.2.5

c) What is the use of Registry?

Ans: Refer to Section 3.2

d) Define Anonymous function.

Ans: Refer to Section 2.2.2

e) Explain Web Server.

Ans: Refer to Section 4.1

f) What is Package? JS on file.

Ans: Refer to Section 3.3

g) Explain global package.

Ans: Refer to Section 3.6

h) List out the parameters of CreateConnection ( ).

Ans: **Options** *<Object>:* It is any <Object> or JSON data which contains connection details.

Callback *<Function>*: It is a callback<function> which receives the created socket.

i) Write down the Syntax to Concatenate Node buffers to a Single Node Buffer.

Ans: Refer to Section 2.3.5

j) Write down the types of modules in Node. JS.

Ans: Refer to Section 2.5

2. Answer the following (Any Four) : **[4 × 4 = 16]**

a) Explain Node. JS Process Model.

Ans: Refer to Section 1.5

b) Write down the steps to create local module.

Ans: Refer to Section 2.5.2

c) Write Node.JS program which will convert the output "SY BCA" into upper-case.

Ans: Refer to Program 2.7 of chapter 2

d) Explain the use of Buffer and How to create Buffer.

Ans: Refer to Sections 2.3.1 and 2.3.2

e) Explain the need of NPM.

Ans: Refer to Section 3.2.1

3. Answer the following (Any Four) : **[4 × 4 = 16]**

a) Write a Node. JS Script to check a given number is Perfect or Not using function.

Ans: Refer to Program 2.10 of chapter 2

b) What is Module? Explain Third Party Module.

Ans: Refer to Sections 2.4 and 2.5.3

c) What is Synchronous and Asynchronous approach?

Ans: Refer to Section 5.1.1

d) What is Event Driven Programming?

Ans: Refer to Section 6.1.1

e) Explain Anonymous function with an example.

Ans: Refer to Section 2.2.2

4. Answer the following (Any Four) : **[4 × 4 = 16]**

a) Explain the syntax to create a text file and delete the file.

Ans: Refer to Section 5.2

b) Explain any two methods of EventEmitter class.

Ans: Refer to Section 6.2

c) Write a Node. JS Script to check a given number is even or odd using function.

Ans: Refer to Program 2.8 of chapter 2

d) Write Node.JS program to count the occurrence of given word in a file and display the count on Console.

Ans:

```
const fs = require('fs');

// Replace 'filename.txt' with the actual path to your file
const filePath = 'filename.txt';
const targetWord = 'word'; // Replace 'word' with the word you want
to count

fs.readFile(filePath, 'utf-8', (err, data) => {
  if (err) {
    console.error('Error reading file:', err);
    return;
  }
```

```
const wordRegex = new RegExp(`\\b${targetWord}\\b`, 'gi');
const matches = data.match(wordRegex);

if (matches) {
  const count = matches.length;
  console.log(`The word "${targetWord}" appears ${count} times in
the file.`);
  } else {
  console.log(`The word "${targetWord}" was not found in the
file.`);
  }
});
```

e) Write a Program to define Module Circle. JS which exports the functions area () and Circumference () and display the details on console.

Ans: Refer to Section 2.5.2

5. Answer the following (Any Two) : **[2 × 3 = 6]**

a) Which databases does Node. JS Supports.

Ans: Node.js being a versatile and flexible runtime environment, can work with various databases. Its ability to interact with databases is mainly facilitated through libraries and packages. Here are some of the popular databases that Node.js supports:

- **MongoDB:** Node.js has a strong integration with MongoDB, a NoSQL database. The official MongoDB Node.js driver allows you to interact with MongoDB databases using asynchronous operations.
- **MySQL:** Node.js can work with MySQL databases using libraries like mysql or mysql2, which provide APIs for executing SQL queries and managing connections.
- **PostgreSQL:** PostgreSQL databases can be accessed from Node.js using libraries like pg or pg-promise. These libraries allow you to perform SQL operations on PostgreSQL databases.
- **SQLite:** SQLite is a lightweight, file-based database that can be accessed directly using the built-in sqlite3 package in Node.js. This makes it convenient for smaller applications or testing scenarios.
- **Redis:** Redis is an in-memory data store often used for caching and quick data retrieval. The redis package enables Node.js applications to interact with Redis databases.
- **Cassandra:** Node.js can connect to Apache Cassandra databases using the cassandra-driver package, which provides tools for working with Cassandra's Query Language (CQL).

- **Elasticsearch:** Elasticsearch, a search and analytics engine, can be integrated with Node.js applications using the elasticsearch package, allowing you to perform advanced search and data analysis.
- **Couchbase:** Couchbase databases can be accessed using the couchbase package in Node.js. Couchbase is a NoSQL database that offers key-value and document storage.
- **Firebase:** Firebase is a backend-as-a-service platform by Google that provides various services including a real-time database. Node.js can interact with Firebase using the firebase-admin package.
- **Microsoft SQL Server:** The mssql package enables Node.js applications to interact with Microsoft SQL Server databases, allowing execution of SQL queries and management of connections.
- **Neo4j:** Neo4j, a graph database, can be accessed using the neo4j-driver package. This enables Node.js applications to work with graph data and perform graph-related operations.

These are just a few examples of databases that Node.js can work with. The Node.js ecosystem is rich with packages and libraries, so you can likely find support for other databases as well. Always check the documentation and community resources for the latest information and best practices when working with a specific database in Node.js.

b) Write down the Connection String of Node.JS and MySQL.

Ans: Refer to Section 7.2

c) Explain the use of REPL.

Ans: Refer to Section 1.10

❖❖❖

WINTER 2022

1. Answer the following (any Eight): **[8 × 2 = 16]**

(a) What is the command to initialize node package manager (NPM)? Write it's syntax.

Ans.: Refer to Section 3.3

(b) What is REPL?

Ans.: Refer to Section 1.10

(c) List any four core modules of node JS.

Ans.: Refer to Section 2.5.1

(d) List any two methods included under path module of node JS.

Ans.: Refer to Section 2.5.1

(e) For which tasks a File System module is used for?

Ans.: Refer to Section 5.2.1

(f) Write a command to add dependency "express" using NPM.

Ans.: Refer to Section 3.8

(g) Write a command to install MYSOL Package by using NPM.

Ans.: `$ npm install mysql`

(h) Write down steps to handle http requests while creating web server using node JS.

Ans.: Refer to Section 4.3

(i) Write any two advantages of node JS.

Ans.: Refer to Section 1.3

(j) Write any two functions of Buffer used in node JS.

Ans.: Refer to Section 2.3

2. Attempt the following (any Four): **[4 × 4 = 16]**

(a) Write a Program to update table records using node JS and MySQL database.

Ans.: Refer to Program 7.6 of Chapter 7

(b) Explain Node JS Process Model with the help of diagram.

Ans.: Refer to Section 1.5

(c) How does Node JS handles a file request?

Ans.: Refer to Section 5.2

(d) What is the Purpose of object module experts in node JS?

Ans.: Refer to Section 2.6

(e) Explain sf.readfile( ) method for all possible ralves of options.

Ans.: Refer to Section 5.2.1

3. Answer the following (any Four): **[4 × 4 = 16]**

(a) Write a Program which uses addlistener( ) method of Event Emmitter class.

Ans.: Refer to Section 6.2

(b) Write a short note on NPM.

Ans.: Refer to Section 3.2

(c) Write a program to delete table records using node JS and MySQL database.

Ans.: Refer to Program 7.7 of Chapter 7

(d) How do you install Packages locally using NPM. Explain with an example.

Ans.: Refer to Section 3.4

(e) Compare Traditional web, server model and Node JS Process model.

Ans.: Refer to Sections 1.4 and 1.5

4. Answer the following (any Four): **[3 × 4 = 12]**

(a) Write a program to use SoL SELECT very to show data from a table using node JS and MySoL database.

Ans.: Refer to Program 7.4 of Chapter 7

(b) Explain steps to install Node JS on windows.

Ans.: Refer to Section 1.8

(c) Write a Program to write a file in node JS.

Ans.: Refer to Program 5.3 of Chapter 5

(d) How to add dependency into Package JS on?

Ans.: Refer to Section 3.5

(d) Write a program to calcolate factorail of given number using function.

Ans.: Refer to Program 2.11 of Chapter 2

5. Attempt the following (any three): **[2 × 3 = 6]**

(a) Explain the meaning, purpose, steps to execute and output of below program:

```
Var http = require ("http");
http.create server (function(req, res){
res. Write head (200, 'content-Type' : 'text/htm'});
res.end('Hellow world'):
})listen (8080);
```

Ans.: Refer to Section 4.2

(b) Explain working of writeHead()

Ans.: Refer to Section 4.2

(c) Explain Inherting events with suitable example.

Ans.: Refer to Section 6.4

❖❖❖

SUMMER 2023

1. Answer the following (Any eight) **[8 × 2 = 16]**

a) What is the Command to initialize Node Package Manager (NPM)? Write it's syntax.

Ans: Refer to Section 3.3

b) What is REPL?

Ans: Refer to Section 1.10

c) List any four core modules of node.JS

Ans: Refer to Section 2.5.1

d) Which directive is used to import node.JS modules?

Ans: Refer to Section 1.9

e) List any 4 methods included under path module of node.JS?

Ans: Refer to Section 2.5.1

f) For which tasks a file system module is used for?

Ans: Refer to Section 5.2.1

g) Write a command to add dependency "express" using NPM.

Ans: Refer to Section 3.8

h) Write a command to install MYSQL package by using NPM.

Ans: `$ npm install mysql`

i) In which situation node. JS is not recommended to use?

Ans: It is not ideal for CPU-intensive tasks.

j) Write steps to handle http requests while creating web server using node.JS?

Ans: Refer to Section 4.3

2. Answer the following (Any Four) **[4 × 4 = 16]**

a) What are the advantages of nodes.JS?

Ans: Refer to Section 1.3

b) Write a program to update table records using node. JS and MYSQL database.

Ans: Refer to Program 7.6 of Chapter 7

c) Explain node.JS process model with the help of diagram.

Ans: Refer to Section 1.5

d) How does node.JS handles a file request?

Ans: Refer to Section 5.2

e) What is the purpose of object module.experts in node.JS?

Ans: Refer to Section 2.6

3. Answer the following (Any four) **[4 × 4 = 16]**

a) Explain fs.readfile( ) method for all possible values of options?

Ans: Refer to Section 5.2.1

b) Write a program which uses addlistener ( ) method of Event Emmitter class.

Ans: Refer to Section 6.2

c) Write a short note on NPM.

Ans: Refer to Section 3.2

d) Create a node.JS file that select all records from the "Customers" table.

Ans: Refer to Program 7.4 of Chapter 7

e) Using node.JS create a web page to read two file names from user and combine in third file.

Ans: Refer to Section 4.2

4. Answer the following (Any four) **[4 × 4 = 16]**

a) What are different features node. JS?

Ans: Refer to Section 1.2.4

b) Compare Traditional web server model and node.JS process model.

Ans: Refer to Sections 1.4 and 1.5

c) Write a program to use SQL SELECT Query to show data from a table using node. JS and MYSQL data base.

Ans: Refer to Program 7.4 of Chapter 7

d) Explain steps to install node. JS on windows.

Ans: Refer to Section 1.8

e) Write a program to write to a file in node.JS.

Ans: Refer to Program 5.3 of Chapter 5

5. Answer the following (Any Two) **[2 × 3 = 6]**

a) Write down the connection string of node.JS and MYSQL

Ans: Refer to Section 7.2

b) Explain Event Driven Programming?

Ans: Refer to Section 6.1.1

c) Explain Anonymous function with an example.

Ans: Refer to Section 2.2.2