

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 11

**Aim : Demonstrate the nodeJS application with DB2 Database.**

Anurag is selling clothes online to his customers via his website. He wants to manage his customer's records and for that he wants to perform following CRUD operations using DB2 and nodeJS :

Practical 11.1: Create a table for storing customers records

Practical 11.2: Add new customer in Customers with (first name, last name, email id, phone number).

Practical 11.3: Add bulk of customers at same time.

Practical 11.4: Display details of all customers.

Practical 11.5: Only display those customer details whose last name="patel"

Practical 11.6: Delete a particular customers record

Practical 11.7: Drop an entire column in a table if not required

Practical 11.8: Delete an entire table.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

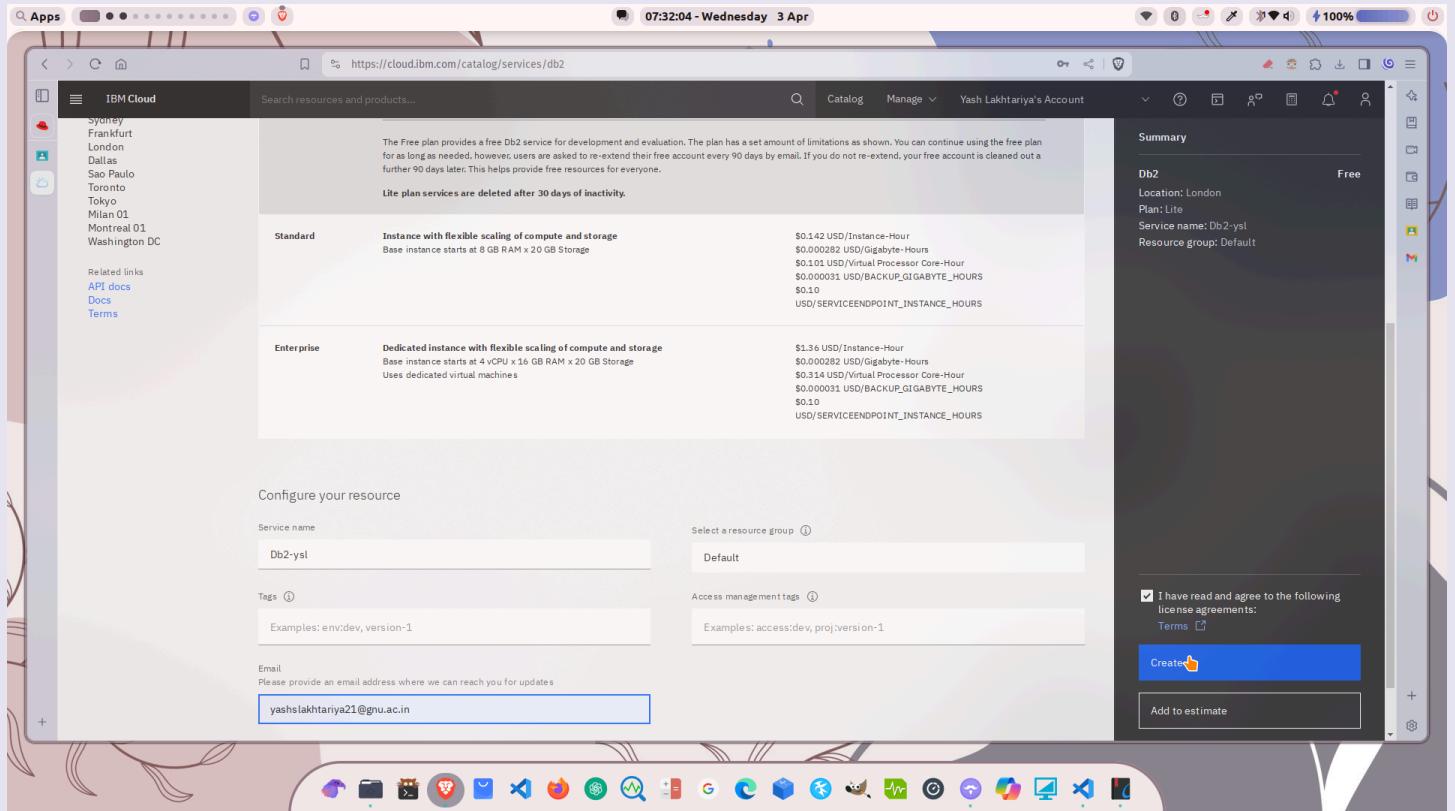
## Steps and Screenshots :

### 1. Create Db2 service on IBM Cloud

The screenshot shows the IBM Cloud Catalog interface for creating a Db2 service. The URL in the browser is <https://cloud.ibm.com/catalog/services/db2>. The page displays the Db2 service details, including its description as a "next generation SQL database. Formerly dashDB For Transactions." The "Create" tab is selected. A dropdown menu for "Select a location" shows "London (eu-gb)" as the chosen option. Below this, under "Select a pricing plan", it says "Displayed prices do not include tax. Monthly prices shown are for country or location: United States". A table compares two plans: "Lite" and "Standard". The "Standard" plan is highlighted with a tooltip: "Instance with flexible scaling of compute and storage. Base instance starts at 8 GB RAM x 20 GB Storage". The "Pricing" section for the Standard plan lists rates: \$0.142 USD/Instance-Hour, \$0.009282 USD/Gigabyte-Hours, \$0.101 USD/Virtual Processor Core-Hour, \$0.00031 USD/BACKUP\_GIGABYTE\_HOURS, and \$0.10 USD/SERVICEENDPOINT\_INSTANCE\_HOURS. A note below states: "The starting configuration provides one SQL database per service instance residing on shared compute slices, with 2 shareable vCPUs (8 GB of memory), and 20 GB of storage for data and logs. All database deployed across multi-tenant compute infrastructure. Scale your database up to 16 vCPUs (64 GB of memory) and 4 TB of storage for data and logs. Standard offers a high availability option that includes one database running on three shared virtual servers. Each HA node is billed separately. The high availability virtual servers are provisioned across multiple availability zones in IBM Cloud regions that support it. Scale your compute and storage independently to achieve the perfect price/performance fit. Each plan includes up to 100 GB of backup storage, stored for 14 days." On the right side, there is a "Summary" panel with fields for "Db2", "Location: London", "Plan: Standard", "Service name: Db2-ysh", and "Resource group: Default". A message box states: "This paid plan cannot be added to an IBM Cloud trial account. You can add a credit card to create a Pay-As-You-Go account. If a free plan for this service is available, you can choose to add it." At the bottom, there is a checkbox for "I have read and agree to the following license agreements:" followed by a "Terms" link, an "Upgrade" button, and a "Add to estimate" button.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 11**

## 2. Assign the preferred name and provide email



The screenshot shows the IBM Cloud catalog interface for creating a Db2 service instance. The URL in the address bar is <https://cloud.ibm.com/catalog/services/db>. The page displays two plan options: 'Standard' and 'Enterprise'. The 'Standard' plan is selected, showing details like 'Instance with flexible scaling of compute and storage' and pricing starting at \$0.442 USD/Instance-Hour. The 'Enterprise' plan is also listed with similar details. On the right side, a summary panel shows the service is 'Free', located in London, Plan: Lite, Service name: Db2-ysl, and Resource group: Default. Below the plans, a configuration section titled 'Configure your resource' allows setting the service name (Db2-ysl), selecting a resource group (Default), adding tags (Examples: env:dev, version-1), and access management tags (Examples: access:dev, proj:version-1). An email field is populated with 'yashslakhtariya21@gnu.ac.in'. A checkbox for accepting license agreements is checked, and a large blue 'Create' button is visible.

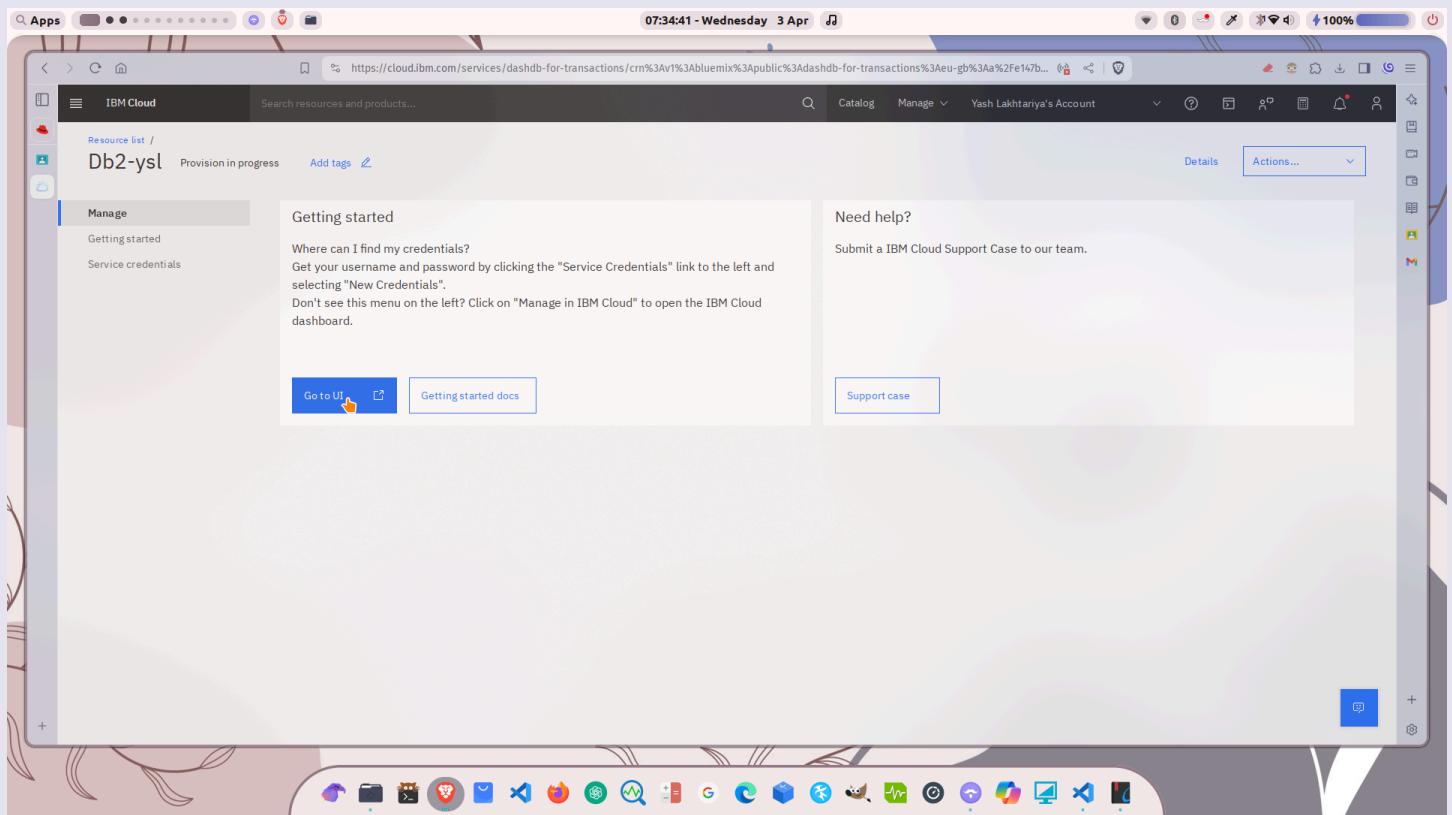
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

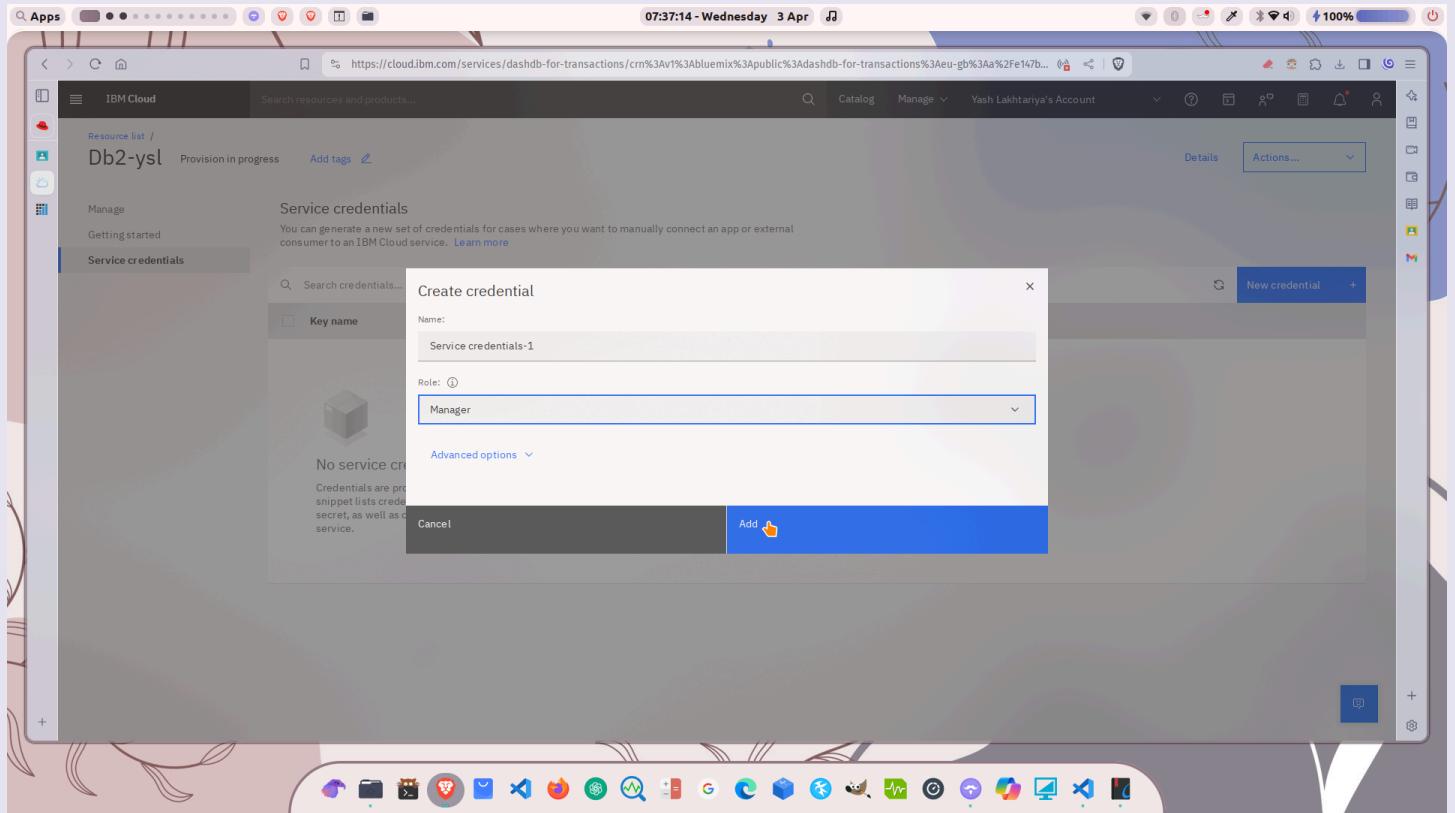
EADC Practical 11

### 3. The service is created and can be visited Db2 using option Go to UI



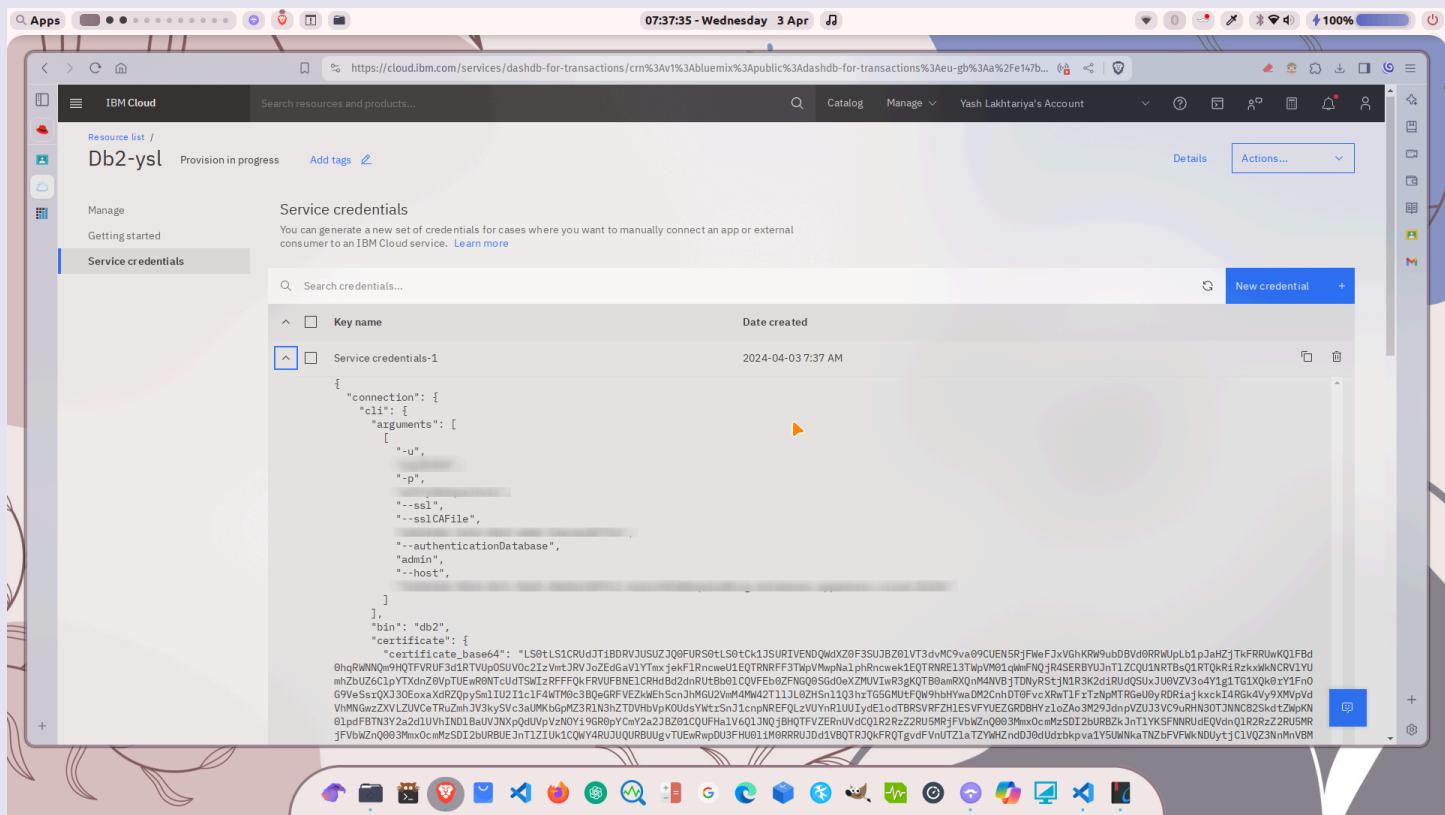
**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 11**

#### 4. First, create the service credentials of Manager Role



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 11**

## 5. Copy the hostname, username and password from it



The screenshot shows the IBM Cloud interface for managing service credentials. The URL in the address bar is <https://cloud.ibm.com/services/dashdb-for-transactions/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2Fe147b...>. The page title is "IBM Cloud" and the sub-page title is "Db2-ysl Provision in progress". The "Service credentials" tab is selected. A search bar at the top right contains the placeholder "Search credentials...". Below the search bar, there is a table with one row visible, showing a credential named "Service credentials-1" created on "2024-04-03 7:37 AM". The credential details are partially redacted. At the bottom of the page, there is a horizontal toolbar with various icons for navigating between different services and applications.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 11

6. Paste the credentials copied in nodejs project file with cloudant connectivity and rebuild the module using **npm rebuild ibm\_db**

The screenshot shows a Mac OS X desktop environment. In the center is a code editor window for VS Code, displaying a file named 'testsql.js'. The code in the editor is as follows:

```
1  db2 > ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelv1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err, conn) {
2     var ibmdb = require("ibm_db");
3
4     // Create Table
5     ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelv1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err, conn) {
6         if (err) return console.log(err);
7
8         conn.query("CREATE TABLE yslcust( f_name VARCHAR(20), l_name VARCHAR(25), email_id VARCHAR(40), mobile_number VARCHAR(20))", function (err, data) {
9             if (err) console.log(err);
10            console.log(data);
11            conn.close(function () {
12                if (err) return console.log(err);
13            });
14        });
15    });
}
```

Below the code editor is a terminal window showing the command 'npm rebuild ibm\_db' being run and completed successfully.

```
yash ~/Practical_11 % main % v18.18.2 07:41 npm rebuild ibm_db
rebuilt dependencies successfully
yash ~/Practical_11 % main % v18.18.2 07:42 npm rebuild ibm_db
yash ~/Practical_11 % main % v18.18.2 07:44
```

### Combined code of all operations :

```
// var ibmdb = require("ibm_db");

// // Create Table
//
ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelv1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err, conn) {

    if (err) return console.log(err);
    conn.query("CREATE TABLE yslcust( f_name VARCHAR(20), l_name
```

**Name - Yash Lakhtariya**

**Enrollment number - 21162101012**

**Branch - CBA      Batch - 61**

**EADC Practical 11**

```
VARCHAR(25), email_id VARCHAR(40), mobile_number VARCHAR(20))", function
(err, data) {

    // if (err) console.log(err);

    // console.log(data);

    // conn.close(function () {
    //     console.log('done');
    // });

    // });

    // });

//////INSERT SINGLE

// var ibmdb = require("ibm_db");

//
ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2
io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelV
v1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err,
conn) {

    // if (err) return console.log(err);

    // conn.query("INSERT INTO wqy86844.yslcust(f_name, l_name, email_id,
mobile_number) VALUES ('Ram', 'Raghav', 'sriram@ayodhya.dham', '9876543210')",
function (err, data) {

        // if (err) console.log(err);
```

**Name - Yash Lakhtariya**

**Enrollment number - 21162101012**

**Branch - CBA      Batch - 61**

**EADC Practical 11**

```
//      console.log(data);

//      conn.close(function () {
//          console.log('done');
//      });

// });
// });

// //Insert multiple

// var ibmdb = require("ibm_db");

//
ibmdb.open ("DRIVER={DB2} ;HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2
io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelV
v1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err,
conn) {

// if (err) return console.log(err);

// conn.query("INSERT INTO wqy86844.yslcust (f_name, l_name, email_id,
mobile_number)
VALUES('Krishna', 'Yadav', 'krishna@vrindavan.dham', '1234567890'), ('Narsimha
', 'Dev', 'narsimha@simhachalam.dham', '1478523690')", function (err, data) {

// if (err) console.log(err);

// console.log(data);
```

**Name - Yash Lakhtariya**

**Enrollment number - 21162101012**

**Branch - CBA      Batch - 61**

**EADC Practical 11**

```
//      conn.close(function () {
//        console.log('done');
//      });

//  });

// });

////Search by filter
// var ibmdb = require("ibm_db");

//
ibmdb.open ("DRIVER={DB2} ;HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2
io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelv
v1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err,
conn) {

//  if (err) return console.log(err);

//  conn.query("SELECT * FROM wqy86844.yslcust WHERE f_name='Ram'", 
function (err, data) {

//  if (err) console.log(err);

//  console.log(data);

//  conn.close(function () {
//    console.log('done');
//  });

//  });

//});
```

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 11**

```
// });

////update any value

// var ibmdb = require("ibm_db");

//


ibmdb.open ("DRIVER={DB2} ;HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2
io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelV
v1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err,
conn) {

// if (err) return console.log(err);

// conn.query("UPDATE wqy86844.yslcust SET f_name= 'SriRam',
mobile_number= '7894561230' WHERE l_name='Raghav'", function (err, data) {

// if (err) console.log(err);

// console.log(data);

// conn.close(function () {
//     console.log('done');
// });

// });

// });

//////////drop a particular reading

// var ibmdb = require("ibm_db");
```

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 11**

```
//  
ibmdb.open ("DRIVER={DB2} ;HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2  
io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelV  
v1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err,  
conn) {  
  
//    if (err) return console.log(err);  
  
//    conn.query("DELETE FROM wqy86844.yslcust WHERE f_name='Narsimha'",  
function (err, data) {  
  
//        if (err) console.log(err);  
  
//        console.log(data);  
  
//        conn.close(function () {  
//            console.log('done');  
//        });  
  
//    });  
  
//});  
  
///////////drop particular column  
  
// var ibmdb = require("ibm_db");  
  
//  
ibmdb.open ("DRIVER={DB2} ;HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2
```

**Name - Yash Lakhtariya**

**Enrollment number - 21162101012**

**Branch - CBA      Batch - 61**

**EADC Practical 11**

```
io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelV
v1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err,
conn) {

//    if (err) return console.log(err);

//    conn.query("ALTER TABLE wqy86844.yslcust DROP COLUMN mobile_number",
function (err, data) {

//        if (err) console.log(err);

//        console.log(data);

//        conn.close(function () {
//            console.log('done');
//        });
//    });

// });

//////////drop table

// var ibmdb = require("ibm_db");

//


ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2
io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelV
v1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err,
conn) {
```

**Name - Yash Lakhtariya**

**Enrollment number - 21162101012**

**Branch - CBA      Batch - 61**

**EADC Practical 11**

```
//    if (err) return console.log(err);

//    conn.query("DROP TABLE wqy86844.yslcust", function (err, data) {

//        if (err) console.log(err);

//        console.log(data);

//        conn.close(function () {
//            console.log('done');
//        });

//    });

//});
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

## 7. Create the table first

A screenshot of a Mac OS X desktop environment. In the foreground, a terminal window is open with the command `node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/testsql.js"`. The output shows the command was run at 07:44 on Wednesday, April 3rd, and completed successfully with the message "[done]". In the background, a code editor (Visual Studio Code) is running. The file `testsql.js` is open, containing the following code:

```
1 var ibmdb = require("ibm_db");
2
3 // Create Table
4 ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8lcg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9T", function (err, conn) {
5   if (err) return console.log(err);
6
7   conn.query("CREATE TABLE yslcust( f_name VARCHAR(20), l_name VARCHAR(25), email_id VARCHAR(40), mobile_number VARCHAR(20))", function (err, data) {
8     if (err) console.log(err);
9
10    console.log(data);
11
12    conn.close(function () {
13      if (err) return console.log(err);
14
15    });
16  });
17});
```

A screenshot of the IBM Db2 Cloud interface. The left sidebar shows navigation options: Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The main area is titled "Tables" and shows a table named "YSLCUST" under the schema "WQY86844". On the right, a "Table definition" panel is open, showing the structure of the "YSLCUST" table:

Name	Data type	Nullable	Length	Scale
F_NAME	VARCHAR	Y	20	0
L_NAME	VARCHAR	Y	25	0
EMAIL_ID	VARCHAR	Y	40	0
MOBILE_NUMBER	VARCHAR	Y	20	0

At the bottom of the table definition panel, there is a blue "View data" button.

**Name - Yash Lakhtariya**

Enrollment number - 21162101012

**Branch - CBA**      **Batch - 61**

# EADC Practical 11

## 8. Insert a single data row in it

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

**Terminal:**

```
yash ./Practical_11 main ② @ v18.18.2 07:47 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/testsql.js"
[]
done
```

**Code Editor (VS Code):**

```
testsql.js U
db2> ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9T...") callback
30
31 var ibmdb = require("ibm_db");
32
33 ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9T...
34
35 if (err) return console.log(err);
36
37 conn.query("INSERT INTO wqy86844.yqlcust(f_name, l_name, email_id, mobile_number) VALUES('Ram','Raghav','sriram@ayodhya.dham','0876543210
38
39 if (err) console.log(err);
40
41 console.log(data);
42
43 conn.close(function () {
44   console.log('done');
45 })
```

**Bottom Status Bar:**

Ln 47, Col 6 Spaces: 2 UTF-8 CRLF () JavaScript Prettier

The screenshot shows the IBM Db2 on Cloud interface. The top navigation bar includes links for Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, Application objects, and a Back button. The main content area displays a table titled "WQY86844.YSLCUST" with the following data:

F_NAME	I_NAME	EMAIL_ID	MOBILE_NUMBER
Ram	Raghav	sriram@ayodhya.dham	9876543210

On the left side, there is a sidebar with various icons for navigation and management, including a magnifying glass for search, a gear for settings, and a plus sign for new items. The bottom of the screen features a decorative floral pattern and a row of small application icons.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

## 9. Insert multiple rows at once

The screenshot shows a macOS desktop environment. On the left, there is a window titled "IBM Db2 on Cloud" displaying a table named "WQY86844.YSLCUST". The table has columns: F\_NAME, L\_NAME, EMAIL\_ID, and MOBILE\_NUMBER. It contains three rows of data:

F_NAME	L_NAME	EMAIL_ID	MOBILE_NUMBER
Krishna	Yadav	krishna@vrindavan.dham	1234567890
Narsimha	Dev	narsimha@simhachalam.dham	1478523690
Ram	Raghav	sriram@ayodhya.dham	9876543210

On the right, there is a terminal window titled "testsq.js" showing the following code:

```
db2 > testsq.js <
56 //Insert multiple
57
58 var ibmdb = require("ibm_db");
59
60 ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2;port=50000;uid=yash;pwd=yash",function(err,conn){
61
62 if (err) return console.log(err);
63
64 conn.query("INSERT INTO wqy86844.yslcust (f_name, l_name, email_id, mobile_n
65
66
67
68
69
70
71
72
73
74
75 }) ; <- #64-74 conn.query
```

The terminal also shows the command being run and its output:

```
yash > ./Practical_11 > ^Z main x? @ v18.18.2 07:49 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/testsql.js"
[]
done
```

```
yash > ./Practical_11 > ^Z main x? @ v18.18.2 07:50
```

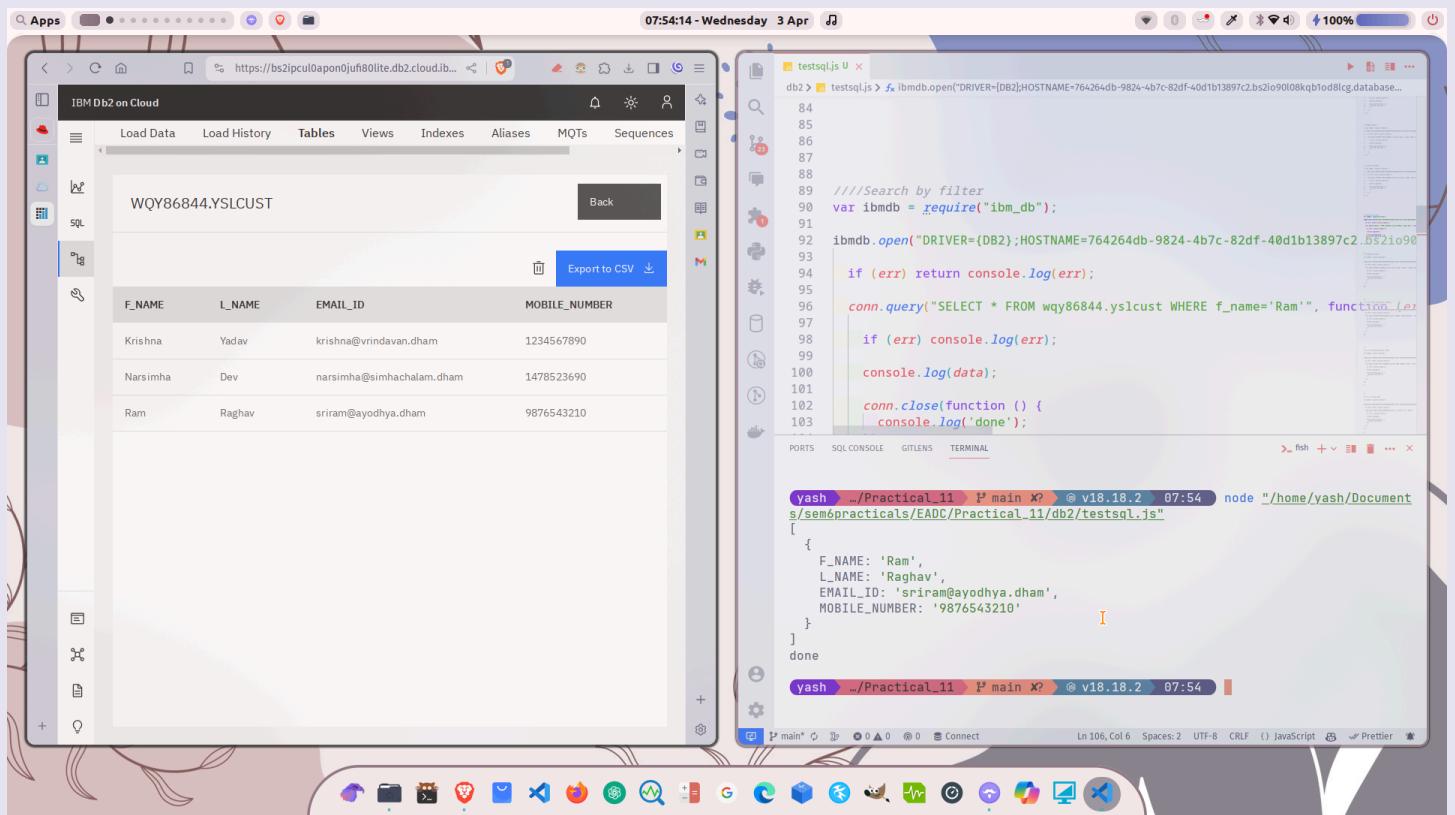
# Name - Yash Lakhtariya

**Enrollment number - 21162101012**

**Branch - CBA**      **Batch - 61**

# EADC Practical 11

## 10. Search for specific row using an attribute



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

## 11. Update details (here phone number and name of 3rd row)

The screenshot shows a Mac desktop with two windows open. On the left is the 'IBM Db2 on Cloud' interface, specifically the 'Tables' section for the 'WQY86844.YSLCUST' table. The table has four columns: F\_NAME, L\_NAME, EMAIL\_ID, and MOBILE\_NUMBER. The data is as follows:

F_NAME	L_NAME	EMAIL_ID	MOBILE_NUMBER
Krishna	Yadav	krishna@vrindavan.dham	1234567890
Narsimha	Dev	narsimha@simhachalam.dham	1478523690
SriRam	Raghav	sriram@ayodhya.dham	7894561230

On the right is a terminal window titled 'testsq.js'. It contains the following JavaScript code using the ibmdb module to update the database:

```
114 //update any value
115 var ibmdb = require("ibm_db");
116
117 ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90
118
119 if (err) return console.log(err);
120
121 conn.query("UPDATE wqy86844.yslcust SET f_name= 'SriRam', mobile_number= 7894561230 WHERE id= 3");
122
123 if (err) console.log(err);
124
125 conn.close(function () {
126   console.log('done');
127 });
128
129
130
131
132
133
134
```

The terminal shows the command run and its output:

```
yash > ./Practical_11 > ^Z main x? @ v18.18.2 07:58 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/testsql.js"
[]
done
yash > ./Practical_11 > ^Z main x? @ v18.18.2 07:58
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

## 12. Delete a specific row (here 2nd)

The screenshot shows a dual-monitor setup. The left monitor displays the IBM Db2 on Cloud interface, specifically the 'Tables' section for the 'WQY86844.YSLCUST' table. The table has two rows:

F_NAME	L_NAME	EMAIL_ID	MOBILE_NUMBER
Krishna	Yadav	krishna@vrindavan.dham	1234567890
SriRam	Raghav	sriram@ayodhya.dham	7894561230

The right monitor shows a terminal window with a Node.js script named 'testsql.js'. The script connects to an IBM DB2 database and performs a DELETE operation on the 'WQY86844.YSLCUST' table where 'f\_name' is 'Narsimha'. The terminal output shows the command being run and the word 'done' indicating the operation was successful.

```
testsql.js ✘
db2> var ibmdb = require("ibm_db");
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.us2lin90
if (err) return console.log(err);
conn.query("DELETE FROM wqy86844.yslcust WHERE f_name='Narsimha'", function
if (err) console.log(err);
console.log(data);
conn.close(function () {
| console.log('done');
});
}); <- #156-166 conn.query
}); <- #152-168 ibmdb.open
PORTS SQL CONSOLE GITLENS TERMINAL
yash ~/Practical_11 ➜ main ✘ @ v18.18.2 07:59 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/testsql.js"
[]
done
yash ~/Practical_11 ➜ main ✘ @ v18.18.2 07:59
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

### 13. Delete the specific column (here mobile number)

The screenshot shows a macOS desktop environment. On the left, there is a window titled "IBM Db2 on Cloud" displaying a table named "WQY86844.YSLCUST". The table has columns F\_NAME, L\_NAME, and EMAIL\_ID, with two rows of data: Krishna/Yadav and SriRam/Raghav. On the right, there is a terminal window titled "testsq.js" showing a Node.js script. The script uses the ibmdb library to connect to a database and execute an ALTER TABLE command to drop the "mobile\_number" column from the "wqy86844.ySLCUST" table. Below the terminal is a system tray with various icons.

```
db2 > testsq.js > f ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b138972.bs2o90l08kqb1od8lcg.database.."
183
184 if (err) return console.log(err);
185
186 conn.query("ALTER TABLE wqy86844.ySLCUST DROP COLUMN mobile_number", function (err, data) {
187   if (err) console.log(err);
188
189   console.log(data);
190
191   conn.close(function () {
192     console.log('done');
193   });
194
195 });
196
197 }); <- #188-198 conn.query
198
199
200 });
201
202
PORTS SQL CONSOLE GITLENS TERMINAL
yash > ./Practical_11 > f main > @ v18.18.2 08:01 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/testsql.js"
[]
done
yash > ./Practical_11 > f main > @ v18.18.2 08:01
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

## 14. Drop the database

The screenshot shows a dual-monitor setup. The left monitor displays the IBM Db2 on Cloud web interface, specifically the 'Tables' section. It shows a single schema named 'WQ...' with one table named 'Jcrl'. A message below the table list says 'You don't have any data currently'. The right monitor displays a terminal window with a Node.js script named 'testsql.js'. The script uses the 'ibmdb' module to connect to a DB2 database and drop a table named 'way86844.yqlcust'. The terminal also shows the command used to run the script: 'node "/home/yash/Documents/sem6practicals/EADC/Practical\_11/db2/testsql.js"'.

```
db2> testsql.js <
210  var ibmdb = require("ibm_db");
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.firebaseio.io:9001", function (err, conn) {
if (err) return console.log(err);
conn.query("DROP TABLE way86844.yqlcust", function (err, data) {
if (err) console.log(err);
console.log(data);
conn.close(function () {
console.log('done');
});
}); <- #218-228 conn.query
#214-230 ibmdb.open
yash > ./Practical_11 > ^Z main x? @ v18.18.2 08:02 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/testsql.js"
[] done
yash > ./Practical_11 > ^Z main x? @ v18.18.2 08:02
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

### TASK :

**Database Setup :** Provision a Db2 instance on IBM Cloud. Create a table named Employees with the following columns :

EmployeeID (integer, primary key)  
FirstName (varchar)  
LastName (varchar)  
Department (varchar)  
Salary (decimal)

### Code :

```
var ibmdb = require("ibm_db");

function createTable(conn) {
    conn.query("CREATE TABLE employees( EmployeeID int primary key not
null, FirstName VARCHAR(25) , LastName VARCHAR(25) , Department VARCHAR(20) ,
Salary decimal(10))",
        function (err, data) {
            if (err) console.log(err);
            console.log(data);
            if (data) console.log('\n\tEmployee Table created
successfully!\n');
        });
}

// add function to get all current details stored in the table
function getEmployees(conn) {
    conn.query("SELECT * FROM employees",
        function (err, data) {
            if (err) console.log(err);
            console.log(data);
        });
}
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

```
function insertEmployee(conn, employee) {
    conn.query(`INSERT INTO employees (EmployeeID, FirstName, LastName,
Department, Salary) VALUES (${employee.EmployeeID},
 '${employee.FirstName}', '${employee.LastName}', '${employee.Department}',
 ${employee.Salary})`,
        function (err, data) {
            if (err) console.log(err);
            console.log(data);
        });
}

function updateSalary(conn, salary, employeeId) {
    conn.query(`UPDATE employees SET Salary = ${salary} WHERE EmployeeID =
${employeeId}`,
        function (err, data) {
            if (err) console.log(err);
            console.log(data);
        });
}

function deleteEmployee(conn, employeeId) {
    conn.query(`DELETE FROM employees WHERE EmployeeID = ${employeeId}`,
        function (err, data) {
            if (err) console.log(err);
            console.log(data);
        });
}

function dropEmployeesTable(conn) {
    conn.query("DROP TABLE employees",
        function (err, data) {
            if (err) console.log(err);
        });
}
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

```
        console.log(data);
    });
}

ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2
io90108kqb1od81cg.databases.appdomain.cloud;UID=wqy86844;PWD=eZ9Ty8b5qzelV
v1z;PORT=32536;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err,
conn) {
    if (err) return console.log(err);

    // createTable(conn);
    // insertEmployee(conn, {EmployeeID: 1, FirstName: 'John', LastName:
'Doe', Department: 'Sales', Salary: 50000});
    // insertEmployee(conn, {EmployeeID: 2, FirstName: 'Jane', LastName:
'Doe', Department: 'Marketing', Salary: 60000});
    // insertEmployee(conn, {EmployeeID: 3, FirstName: 'Jim', LastName:
'Doe', Department: 'HR', Salary: 70000});
    // getEmployees(conn);
    // updateSalary(conn, 70000, 1);
    // deleteEmployee(conn, 3);
    // dropEmployeesTable(conn);
});
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

## Data Operations :

- Create table first :

The screenshot shows a desktop environment with two main windows open. On the left is the 'IBM Db2 on Cloud' interface, which includes a sidebar with icons for Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, and Sequences. The main area displays 'Schemas' and 'Tables'. Under 'Schemas', there is one entry: 'WQ...' (User). Under 'Tables', there are two entries: 'EMPLO...' (Schema WQY86844) and 'ORDER' (Schema WQY86844). A 'New table' button is visible at the top of the table list. At the bottom of the interface, it says 'Total: 1, selected: 1' and 'Total: 2, selected: 0'. On the right is a terminal window titled 'tasksql.js'. It contains the following code:

```
function dropEmployeesTable(conn) {
    conn.query("DROP TABLE employees",
        function (err, data) {
            if (err) console.log(err);
            console.log(data);
        });
}

ibmdb.open("DRIVER=(DB2);HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2ic90",
    if (err) return console.log(err);

createTable(conn);
// insertEmployee(conn, {EmployeeID: 1, FirstName: 'John', LastName: 'Doe'}
// insertEmployee(conn, {EmployeeID: 2, FirstName: 'Jane', LastName: 'Doe'}
// insertEmployee(conn, {EmployeeID: 3, FirstName: 'Jim', LastName: 'Doe'});
// getEmployees(conn);
// updateSalary(conn, 70000, 1);
// deleteEmployee(conn, 3);
dropEmployeesTable(conn);

node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/tasksql.js"
[yash ~/Practical_11 10:49] node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/tasksql.js"
[]

Employee Table created successfully!
```

Below the terminal window, a status bar indicates 'Ln 63, Col 4 Spaces:4 UTF-8 LF () JavaScript Prettier'.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 11

- Insert at least three records into the Employees table with different employee details (EmployeeID, FirstName, LastName, Department, Salary).

The screenshot shows a macOS desktop environment. On the left, the IBM Db2 on Cloud interface is open, displaying the 'EMPLOYEES' table with one record: EmployeeID 1, FirstName John, LastName Doe, Department Sales, and Salary 50000. An 'Export to CSV' button is visible. On the right, a terminal window shows the execution of a Node.js script named 'tasksql.js'. The script performs several database operations: it drops the 'EmployeesTable' if it exists, creates the table, inserts three new employee records (EmployeeID 1, FirstName 'John', LastName 'Doe'; EmployeeID 2, FirstName 'Jane', LastName 'Doe'; EmployeeID 3, FirstName 'Jim', LastName 'Doe'), updates the salary for EmployeeID 1 to 70000, and deletes EmployeeID 3. Finally, it drops the table again. The terminal output indicates that the table was created successfully.

```
10:50:44 - Wednesday 3 Apr 2023 100% ⚡

IBM Db2 on Cloud
Load Data Load History Tables Views Indexes Aliases MQTs Sequences
WQY86844.EMPLOYEES
EMPLOYEEID FIRSTNAME LASTNAME DEPARTMENT SALARY
1 John Doe Sales 50000
Back Export to CSV

tasksql.js tasksq1.js ✘ ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.database... db2 44 function dropEmployeesTable(conn) {
45   conn.query(`DROP TABLE IF EXISTS employees`);
46   console.log(`Table dropped`);
47   conn.query(`CREATE TABLE employees (EmployeeID INT, FirstName VARCHAR(50), LastName VARCHAR(50), Department VARCHAR(50), Salary DECIMAL(10,2));
48   console.log(`Table created`);
49   conn.query(`INSERT INTO employees VALUES (1, 'John', 'Doe', 'Sales', 50000);
50 } <- #44-50 function dropEmployeesTable(conn)
51 ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.database...
52   if (err) return console.log(err);
53
54   // createTable(conn);
55   insertEmployee(conn, {EmployeeID: 1, FirstName: 'John', LastName: 'Doe', Department: 'Sales'});
56   // insertEmployee(conn, {EmployeeID: 2, FirstName: 'Jane', LastName: 'Doe'});
57   // insertEmployee(conn, {EmployeeID: 3, FirstName: 'Jim', LastName: 'Doe'});
58   // getEmployees(conn);
59   // updateSalary(conn, 70000, 1);
60   // deleteEmployee(conn, 3);
61   // dropEmployeesTable(conn);
62 } <- #52-63 ibmdb.open
63
PORTS SQL CONSOLE GITLENS TERMINAL
node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/tasksql.js"
yash ./Practical_11 ▶ main ? @ v18.18.2 10:49 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/tasksql.js"
[]
Employee Table created successfully!

yash ./Practical_11 ▶ main ? @ v18.18.2 10:49 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/tasksql.js"
[]
yash ./Practical_11 ▶ main ? @ v18.18.2 10:50
main* ⚡ 0 0 0 0 Connect Ln 58, Col 27 Spaces:4 UTF-8 LF () JavaScript ⚡ Prettier ⚡
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

The screenshot shows a macOS desktop environment. On the left, the IBM Db2 on Cloud interface is open, displaying the 'EMPLOYEES' table with three rows of data:

EMPLOYEEID	FIRSTNAME	LASTNAME	DEPARTMENT	SALARY
1	John	Doe	Sales	50000
2	Jane	Doe	Marketing	60000
3	Jim	Doe	HR	70000

An 'Export to CSV' button is visible at the top right of the table view. On the right side of the screen, a terminal window is open, showing the execution of a JavaScript file named 'tasksql.js'. The code performs database operations like creating a table, inserting data, and updating salary. The terminal output shows the command run and the successful creation of the 'Employee' table.

```
db2 > ibmdb.open("DRIVER=(DB2);HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.database..")
44  function dropEmployeesTable(conn) {
45    function (err, data) {
46      if (err) console.log(err);
47      else console.log(data);
48    }
49  } <- #44-50 function dropEmployeesTable(conn)
50
51 ibmdb.open("DRIVER=(DB2);HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.database.."
52   if (err) return console.log(err);
53
54   // createTable(conn);
55   // insertEmployee(conn, {EmployeeID: 1, FirstName: 'John', LastName: 'Doe'}
56   insertEmployee(conn, {EmployeeID: 2, FirstName: 'Jane', LastName: 'Doe', Department: 'Marketing', Salary: 60000});
57   insertEmployee(conn, {EmployeeID: 3, FirstName: 'Jim', LastName: 'Doe', Department: 'HR', Salary: 70000});
58
59   // getEmployees(conn);
60   // updateSalary(conn, 70000, 1);
61   // deleteEmployee(conn, 3);
62   // dropEmployeesTable(conn);
63 } <- #52-63 ibmdb.open

PORTS SQL CONSOLE GITLENS TERMINAL
Code + ...
```

```
/sem6practicals/EADC/Practical_11/db2/tasksql.js
[]

Employee Table created successfully!
```

```
yash ~/Practical_11 > main ? @ v18.18.2 : 10:49 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/tasksql.js"
[]
yash ~/Practical_11 > main ? @ v18.18.2 : 10:50 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/tasksql.js"
[]
yash ~/Practical_11 > main ? @ v18.18.2 : 10:51
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 11

- Retrieve all records from the Employees table.

The screenshot shows a dual-monitor setup. The left monitor displays the IBM Db2 on Cloud interface, specifically the 'Tables' section for the 'WQY86844.EMPLOYEES' table. The table has columns: EMPLOYEEID, FIRSTNAME, LASTNAME, DEPARTMENT, and SALARY. It contains three rows of data: (1, John, Doe, Sales, 50000), (2, Jane, Doe, Marketing, 60000), and (3, Jim, Doe, HR, 70000). An 'Export to CSV' button is visible at the top right of the table view. The right monitor displays a terminal window titled 'tasksql.js'. The code in the terminal is a Node.js script that interacts with a DB2 database. It includes functions for creating a table, inserting employees, updating salary, deleting employees, and dropping the table. The terminal shows the execution of the script and its output, which matches the data in the Db2 table.

EMPLOYEEID	FIRSTNAME	LASTNAME	DEPARTMENT	SALARY
1	John	Doe	Sales	50000
2	Jane	Doe	Marketing	60000
3	Jim	Doe	HR	70000

```
1bndb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2;PORT=50000;DATABASE=WQY86844;UID=yash;PWD=yash");
// createTable(conn);
55 // insertEmployee(conn, {EmployeeID: 1, FirstName: 'John', LastName: 'Doe'});
56 // insertEmployee(conn, {EmployeeID: 2, FirstName: 'Jane', LastName: 'Doe'});
57 // insertEmployee(conn, {EmployeeID: 3, FirstName: 'Jim', LastName: 'Doe'});
58 getEmployees(conn);
59 // updateSalary(conn, 70000, 1);
60 // deleteEmployee(conn, 3);
61 // dropEmployeesTable(conn);
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 11

- Update the salary of an employee with EmployeeID 1 to \$70,000.

The screenshot shows a desktop environment with three main windows:

- IBM Db2 on Cloud:** A web-based interface for managing a database. It displays a table titled "WQY86844.EMPLOYEES" with the following data:

EMPLOYEEID	FIRSTNAME	LASTNAME	DEPARTMENT	SALARY
1	John	Doe	Sales	70000
2	Jane	Doe	Marketing	60000
3	Jim	Doe	HR	70000

- Code Editor:** An IDE window showing two files: "testsq.js" and "tasksq.js". The "tasksq.js" file contains the following JavaScript code:

```
ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2;PORT=50000;DATABASE=ibmdb;UID=yash;PWD=yash");
// createTable(conn);
// insertEmployee(conn, {EmployeeID: 1, FirstName: 'John', LastName: 'Doe'});
// insertEmployee(conn, {EmployeeID: 2, FirstName: 'Jane', LastName: 'Doe'});
// insertEmployee(conn, {EmployeeID: 3, FirstName: 'Jim', LastName: 'Doe'});
getEmployees(conn);
updateSalary(conn, 70000, 1);
// deleteEmployee(conn, 3);
// dropEmployeesTable(conn);
}); <- #52-63 ibmdb.open
```

- Terminal:** A terminal window showing the command "node "/home/yash/Documents/sem6practicals/EADC/Practical\_11/db2/tasksq.js"" being run. The output shows the updated salary for employee ID 1:

```
DEPARTMENT: 'Marketing',
SALARY: 60000
},
{
EMPLOYEEID: 3,
FIRSTNAME: 'Jim',
LASTNAME: 'Doe',
DEPARTMENT: 'HR',
SALARY: 70000
}

yash .../Practical_11 ➜ main ? @ v18.18.2 10:51 node "/home/yash/Documents/sem6practicals/EADC/Practical_11/db2/tasksq.js"
[]

yash .../Practical_11 ➜ main ? @ v18.18.2 10:52
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 11

- Delete the record of the employee with EmployeeID 3.

The screenshot shows a dual-monitor setup. The left monitor displays the IBM Db2 on Cloud interface, specifically the 'EMPLOYEES' table for database 'WQY86844'. The table contains two rows of data:

EMPLOYEEID	FIRSTNAME	LASTNAME	DEPARTMENT	SALARY
1	John	Doe	Sales	70000
2	Jane	Doe	Marketing	60000

The right monitor displays a terminal window with a Node.js script named 'tasksol.js' running. The script connects to an IBM DB2 database and performs several operations: creating a table, inserting two rows of data, updating the salary of the first row to 70000, deleting the third row (EmployeeID 3), and finally dropping the employees table. The terminal shows the command 'node ./home/yash/Documents/sem6practicals/EADC/Practical\_11/db2/tasksol.js' being run at 10:52 on a v18.18.2 host.

```
db2> ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2;PORT=50000;DATABASE=WQY86844;UID=yash;PWD=yash");
52 // createTable(conn);
53 // insertEmployee(conn, {EmployeeID: 1, FirstName: 'John', LastName: 'Doe'});
54 // insertEmployee(conn, {EmployeeID: 2, FirstName: 'Jane', LastName: 'Doe'});
55 // insertEmployee(conn, {EmployeeID: 3, FirstName: 'Jim', LastName: 'Doe'});
56 // getEmployees(conn);
57 // updateSalary(conn, 70000, 1);
58 // deleteEmployee(conn, 3);
59 // dropEmployeesTable(conn);
60
61
62
63 }); <- #52-63 ibmdb.open
```

```
yash ~/Practical_11 > node main.js
yash ~/Practical_11 > node ./home/yash/Documents/sem6practicals/EADC/Practical_11/db2/tasksol.js
[]

yash ~/Practical_11 > node main.js
yash ~/Practical_11 >
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 11

**Cleanup : Drop the Employees table from the database.**

The screenshot shows a macOS desktop environment. On the left, the IBM Db2 on Cloud interface is open, displaying a warning message: "HWCADM0002E: Table WQY86844.EMPLOYEES can't be found." Below this, the Schemas and Tables sections are visible. The Schemas section shows one schema named "WQ...". The Tables section is empty, showing a table header with columns "Name", "Schema", and "Properties". On the right, a terminal window is open with a code editor showing a file named "tasksql.js". The code contains several database operations, including creating a table, inserting three employees, updating their salary, deleting them, and finally dropping the "EMPLOYEES" table. The terminal window shows the command "node tasksql.js" being run, and the output indicates the script has completed successfully. The system tray at the bottom shows various application icons.

```
db2 > ibmdb.open("DRIVER={DB2};HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2;bs2o90l08kqb1od8lcg.database..");
52 // createTable(conn);
53 // insertEmployee(conn, {EmployeeID: 1, FirstName: 'John', LastName: 'Doe');
54 // insertEmployee(conn, {EmployeeID: 2, FirstName: 'Jane', LastName: 'Doe');
55 // insertEmployee(conn, {EmployeeID: 3, FirstName: 'Jim', LastName: 'Doe');
56 // getEmployees(conn);
57 // updateSalary(conn, 70000, 1);
58 // deleteEmployee(conn, 3);
59 // dropEmployeesTable(conn);
60 });
61 });
62 });
63 });
64 ibmdb.open();
```

```
yash ~/Practical_11 > node tasksql.js
yash ~/Practical_11 >
```