



**Name: Tushar Panchal**

**En.No: 21162101014**

**Sub: EADC (Enterprise Application Development for Cloud)**

**Branch: CBA**

**Batch:61**

## **PRACTICAL 11**

**❖ Question :**

**Demonstrate the nodeJS application with DB2 Database. Anurag is selling clothes online to his customers via his website. He is 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.

**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)

## Data Operations:

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

Retrieve all records from the Employees table.

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

Delete the record of the employee with EmployeeID 3.

## Cleanup:

Drop the Employees table from the database.

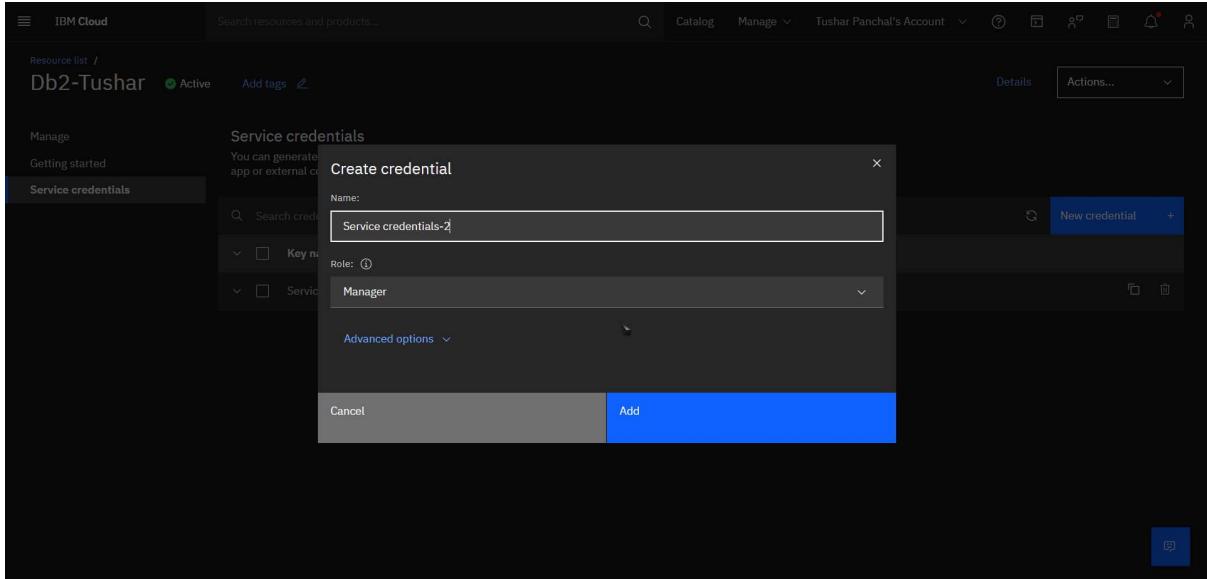
## » Step 1: Navigate DB2 and create service:

The screenshot shows the IBM Cloud Catalog interface for creating a Db2 service. The 'Create' tab is active. The location is set to 'London (eu-gb)'. The plan selected is 'Lite', which includes '200 MB of data storage', '5 simultaneous connections', and 'Shared multitenant system'. A note states that the free plan is deleted after 30 days of inactivity. A checkbox for accepting license agreements is checked. A large blue 'Create' button is prominent at the bottom right.

Name it and hit create after that you will see service created as below:

The screenshot shows the IBM Cloud Resource list for the 'Db2-Tushar' service. The 'Manage' tab is selected. It features a 'Getting started' section with links to 'Go to UI' and 'Getting started docs'. A 'Need help?' section with a 'Support case' button is also present.

## » Step 2: go on to service credentials and create credentials



As you can see below service credentials has been created..

```

{
  "connection": {
    "cli": {
      "arguments": [
        "-u",
        "hmp68104",
        "-p",
        "2bjfiftpdgS6LDmF",
        "-ssl",
        "-sslCAfile",
        "1dd140c-1b52-4f63-a606-53ecba28771d",
        "-authenticationDatabase",
        "admin",
        "-host",
        "824df14d-99de-440d-9991-629c01b3832d.bs2io90108kqb1od8lcg.databases.appdomain.cloud:30119"
      ]
    },
    "bin": "db2"
  }
}
  
```

## » Practical 11.1 : Create a table for storing customers records.

I Downloaded the rar file then faculty provided in classroom

Then make sure you enter cloudant service credentials in code:

```

File Edit Selection View Go Run ...
JS testsqll.js
1 var ibmdb = require("ibm_db");
2
3 // Create Table
4 ibmdb.open("DRIVER={DB2};HOSTNAME=824df4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp68104;PWD=zpjIftp1dgS0LdmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err, conn){
5
6     if (err) return console.log(err);
7
8     conn.query("CREATE TABLE tushar (f_name VARCHAR(20), l_name VARCHAR(25), email_id VARCHAR(40), mobile_number VARCHAR(20))", function (err, data) {
9
10        if (err) console.log(err);
11
12        console.log(data);
13
14        conn.close(function () {
15            console.log('done');
16        });
17    });
18})

```

TERMINAL

```

state: '0B001'
>_pwshe Practical_11 8ms
node .\testsqll.js
[]
done
>_pwshe Practical_11 693ms

```

Ln 10, Col 31 Spaces: 2 UTF-8 CRLF () JavaScript ⚡ Go Live ⚡ Colorize: 0 variables ⚡ Colorize

Then ibm db modules using these commands:

**npm i ibm\_db**

**npm rebuild ibm\_db**

**npm install**

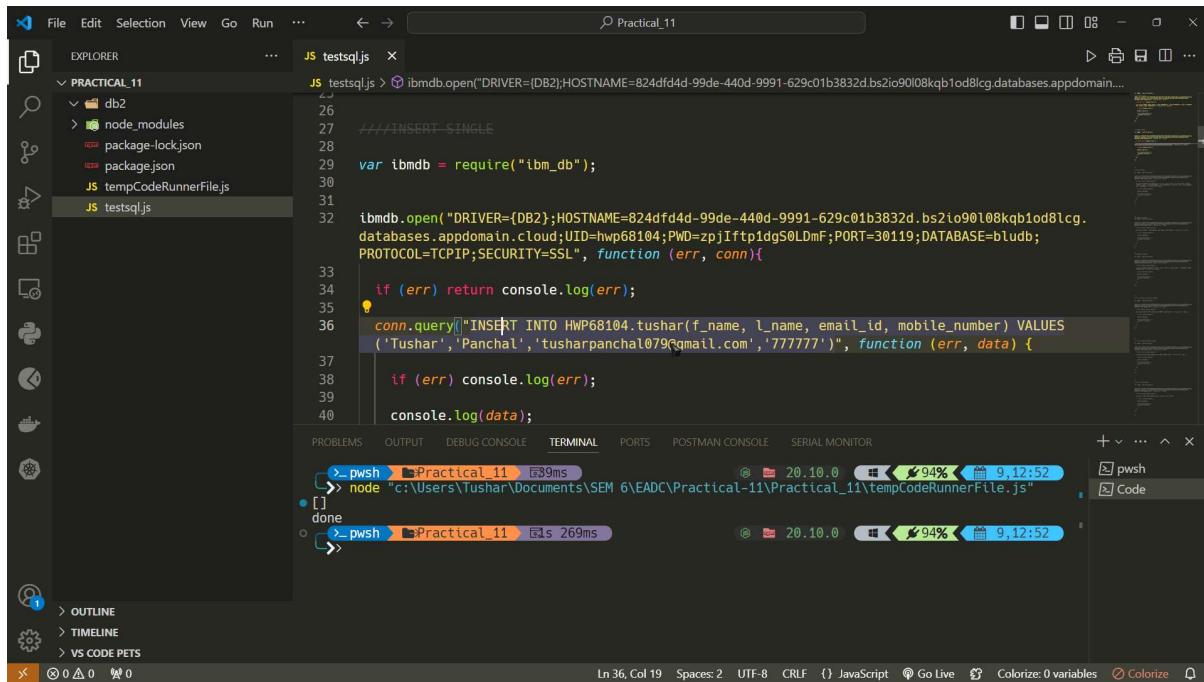
Then run this js code to create table.

We can see that table named tushar has been created below:

Tables		
Name	Schema	Properties
TK	HWP68104	...
TUSHAR	HWP68104	...

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

I inserted a record into the 'tushar' table in IBM Db2 using a Node.js script. The record includes the first name 'tushar', last name 'panchal', email 'tusharpanchal079@gmail.com', and mobile number '777777777' and add schema id before table name



```

File Edit Selection View Go Run ...
JS testsqljs x
PRACTICAL_11
db2
node_modules
package-lock.json
package.json
tempCodeRunnerFile.js
JS testsqljs

26
27 //INSERT SINGLE
28
29 var ibmdb = require("ibm_db");
30
31 ibmdb.open("DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp68104;PWD=zpjIftp1dgS0LDmf;PORT=30119;DATABASE=bludb;PROTOCOL=TCPPIP;SECURITY=SSL", function (err, conn){
32
33 if (err) return console.log(err);
34
35 conn.query("INSERT INTO HWP68104.tushar(f_name, l_name, email_id, mobile_number) VALUES ('Tushar', 'Panchal', 'tusharpanchal079@gmail.com', '777777777')", function (err, data) {
36
37 if (err) console.log(err);
38
39 console.log(data);
40
}

```

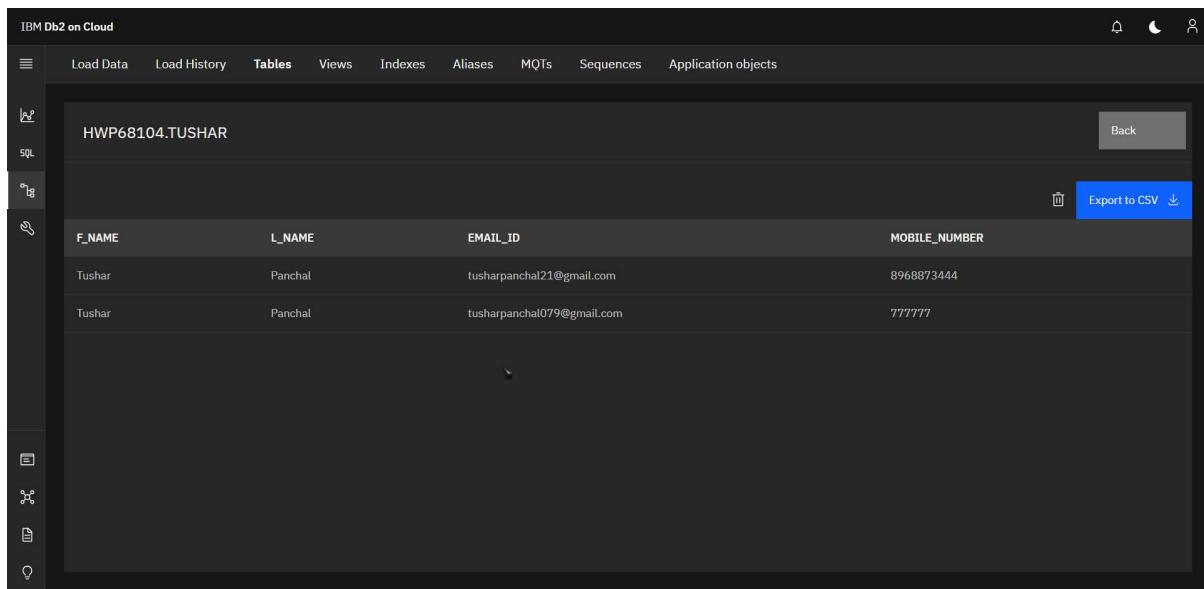
TERMINAL

```

>_psh Practical_11 39ms
> node "c:\Users\tushar\Documents\SEM 6\EADC\Practical-11\Practical_11\tempCodeRunnerFile.js"
[done]
>_psh Practical_11 269ms

```

As you can see below record successfully inserted in table:



F_NAME	L_NAME	EMAIL_ID	MOBILE_NUMBER
Tushar	Panchal	tusharpanchal21@gmail.com	8968873444
Tushar	Panchal	tusharpanchal079@gmail.com	777777777

## ➤ Practical 11.3 : Add bulk of customers at same time.

I successfully inserted multiple records into the 'tushar' table in Db2. The records include entries for 'james bond', 'tom cruise', with corresponding email addresses and mobile numbers.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top bar includes the standard File, Edit, Selection, View, Go, Run, and Terminal tabs. The title bar displays the file name "Practical\_11". The left sidebar features the Explorer, Search, and Problems sections, with the Explorer tab currently active, showing a tree view of the project structure including "PRACTICAL\_11", "db2", "node\_modules", and several JavaScript files like "testsq1.js", "tempCodeRunnerFile.js", and "testsq1.js". The main editor area contains the code for "testsq1.js", which uses the ibm\_db module to insert data into a database. The bottom navigation bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, POSTMAN CONSOLE, and SERIAL MONITOR. The bottom status bar shows the current line (Ln 65, Col 119), character count (Spaces: 2), line endings (CRLF), language (JavaScript), and other status indicators like "Go Live" and "Colorize: 0 variables". A terminal tab at the bottom is titled "pwsh" and shows the command "Practical\_11 >".

```
JS testsq1.js
JS testsq1.js > ibmdb.open("DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp68104;PWD=zpjIftp1dgS0LDmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err, conn){
  if (err) return console.log(err);
  conn.query("INSERT INTO HWP68104.tushar(f_name, l_name, email_id, mobile_number) VALUES ('James', 'Bond', 'jamesbond007@gmail.com', '777777'), ('Tom', 'Cruise', 'tomcruise@gmail.com', '1234567890')", function (err, data) {
    if (err) console.log(err);
    console.log(data);
    conn.close(function () {
      console.log('done');
    });
  });
}
```

We can see the successful insertion of multiple records into the 'tushar' table in the IBM Db2 UI.

IBM Db2 on Cloud

Load Data Load History Tables Views Indexes Aliases MQTs Sequences Application objects

Back

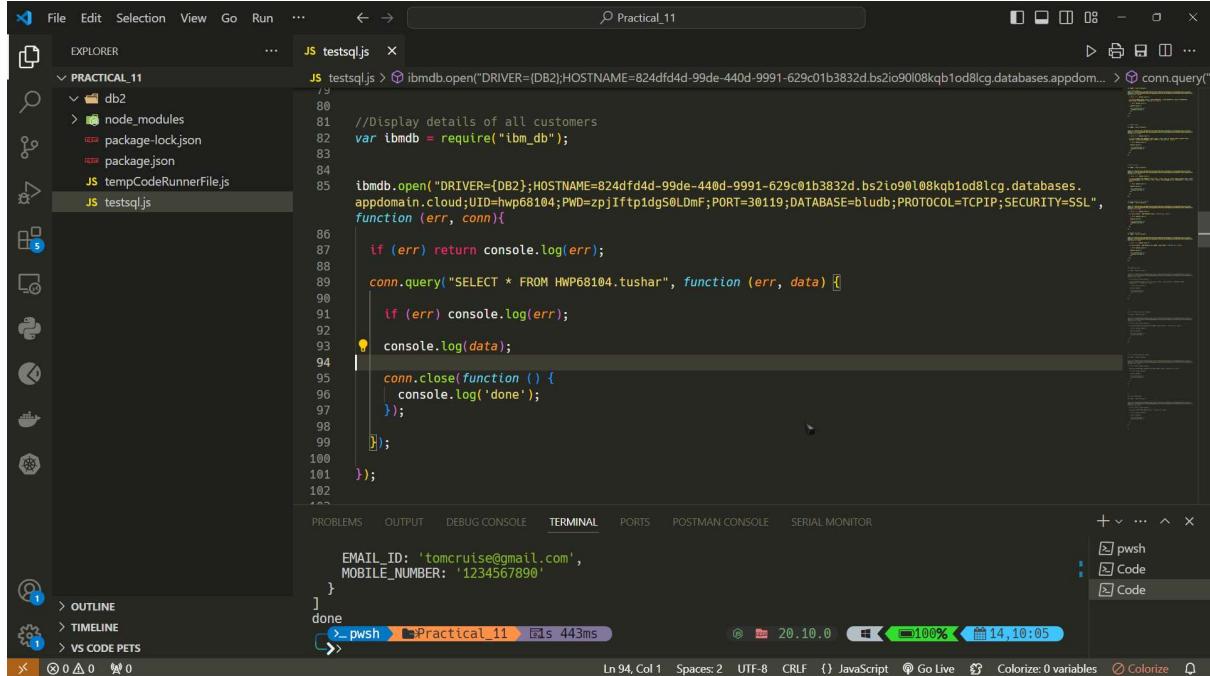
HWP68104.TUSHAR

Export to CSV

F_NAME	L_NAME	EMAIL_ID	MOBILE_NUMBER
James	Bond	jamesbond007@gmail.com	777777
Tom	Cruise	tomcruise@gmail.com	1234567890
Tushar	Panchal	tusharpanchal21@gmail.com	8968873444
Tushar	Panchal	tusharpanchal079@gmail.com	777777

## » Practical 11.4 : Display details of all customers.

I used a Node.js script to query the 'tushar' table in IBM Db2, retrieving all records. The data was successfully fetched and displayed, confirming the search operation in the database.

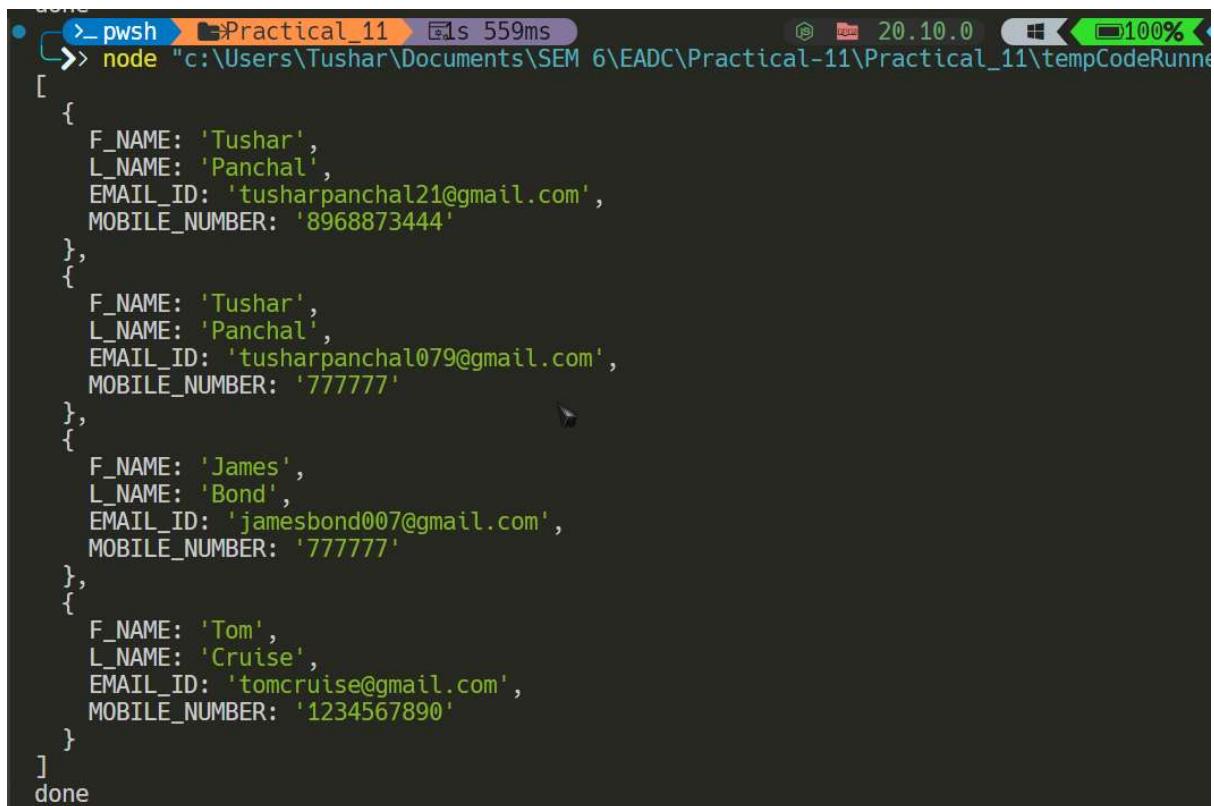


```

File Edit Selection View Go Run ... ← → 🔍 Practical_11
EXPLORER JS testsql.js ...
PRACTICAL_11 ibmdb.open("DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=50119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", conn.query);
node_modules
  package-lock.json
  package.json
JS tempCodeRunnerFile.js
JS testsql.js
ibmdb.open("DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=50119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", conn.query);
if (err) return console.log(err);
conn.query("SELECT * FROM HWP68104.tushar", function (err, data) {
  if (err) console.log(err);
  console.log(data);
  conn.close(function () {
    console.log('done');
  });
});
done
EMAIL_ID: 'tomcruise@gmail.com',
MOBILE_NUMBER: '1234567890'
]
done
20.10.0 100% 14,10:05
Ln 94, Col 1 Spaces: 2 UFT-8 CRLF Go Live Colorize: 0 variables Colorize

```

Here see the output of above question:



```

>_psh >Practical_11 ls 559ms 20.10.0 100%
>> node "c:\Users\Tushar\Documents\SEM 6\EADC\Practical-11\Practical_11\tempCodeRunnerFile.js"
[
  {
    F_NAME: 'Tushar',
    L_NAME: 'Panchal',
    EMAIL_ID: 'tusharpanchal21@gmail.com',
    MOBILE_NUMBER: '8968873444'
  },
  {
    F_NAME: 'Tushar',
    L_NAME: 'Panchal',
    EMAIL_ID: 'tusharpanchal079@gmail.com',
    MOBILE_NUMBER: '777777'
  },
  {
    F_NAME: 'James',
    L_NAME: 'Bond',
    EMAIL_ID: 'jamesbond007@gmail.com',
    MOBILE_NUMBER: '777777'
  },
  {
    F_NAME: 'Tom',
    L_NAME: 'Cruise',
    EMAIL_ID: 'tomcruise@gmail.com',
    MOBILE_NUMBER: '1234567890'
  }
]
done

```

## » **Practical 11.5 : only display those customer details whose last name="panchal".**

I utilized a Node.js script to query the 'tushar' table in IBM Db2, specifically retrieving records where the last name is 'patel'. The data was successfully fetched and displayed, confirming the search operation in the database

```
File Edit Selection View Go Run ... ← → Practical_11
EXPLORER PRACTICAL_11 ...
db2 node_modules package-lock.json package.json tempCodeRunnerFile.js testsql.js
JS testsql.js x
JS testsql.js > ibmdb open("DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2i090l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp08104;PWD=zpjIftp1dg50LdmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL");
103 //Search by filter
104 var ibmdb = require("ibm_db");
105
106 // Establish connection to the database
107 ibmdb.open("DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2i090l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp08104;PWD=zpjIftp1dg50LdmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err, conn) {
108   if (err) return console.log(err);
109
110   // Execute SQL query to retrieve data where last name is 'panchal'
111   conn.query("SELECT * FROM HWP08104.tushar WHERE l_name='Panchal'", function (err, data) {
112     if (err) return console.log(err);
113
114     // Display the retrieved data
115     console.log(data);
116
117     // Close the database connection
118     conn.close(function (err) {
119       if (err) console.error(err);
120       console.log('Connection closed.');
121     });
122   });
123 });
124
125
126
MOBILE_NUMBER: '8968873444'
}
{
  F_NAME: 'Tushar',
  L_NAME: 'Panchal',
  EMAIL_ID: 'tusharpanchal079@gmail.com',
  MOBILE_NUMBER: '777777'
```

## Output :

```
>_ pwsh └─ Practical_11 └─ ls 311ms
20.10.0 100%
>> node "c:\Users\Tushar\Documents\SEM 6\EADC\Practical-11\Practical_11\tempCodeRunnerFile.js"
[ {
  F_NAME: 'Tushar',
  L_NAME: 'Panchal',
  EMAIL_ID: 'tusharpanchal21@gmail.com',
  MOBILE_NUMBER: '8968873444'
},
{
  F_NAME: 'Tushar',
  L_NAME: 'Panchal',
  EMAIL_ID: 'tusharpanchal079@gmail.com',
  MOBILE_NUMBER: '777777'
}
]
Connection closed.
```

## Practical 11.6 : Delete a particular customers record.

I performed a record deletion operation in IBM Db2, specifically removing the entry from the 'tushar' table where the first name was 'Tushar\_2.0'.

```
File Edit Selection View Go Run ... ← → Practical_11
EXPLORER PRACTICAL_11
  db2 node_modules
    package-lock.json
    package.json
JS tempCodeRunnerFile.js JS testsql.js
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
JS testsql.js x
JS testsql.js > ibmdb.open("DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2i090l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp68104;PWD=zpjIftp1dgS0LdmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err, conn){
  if (err) return console.log(err);
  conn.query("DELETE FROM HWP68104.tushar WHERE f_name='Tushar_2.0'", function (err, data) {
    if (err) console.log(err);
    console.log(data);
    conn.close(function () {
      console.log('done');
    });
  });
});
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SERIAL MONITOR
pwsh Practical_11 401ms 20.10.0 100% 14:10:18
node "c:\Users\Tushar\Documents\SEM 6\EADC\Practical-11\Practical_11\tempCodeRunnerFile.js"
[]
done
pwsh Practical_11 267ms 20.10.0 100% 14:10:20
[]
```

As you can see here below record is deleted successfully:

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

HWP68104.TUSHAR Back

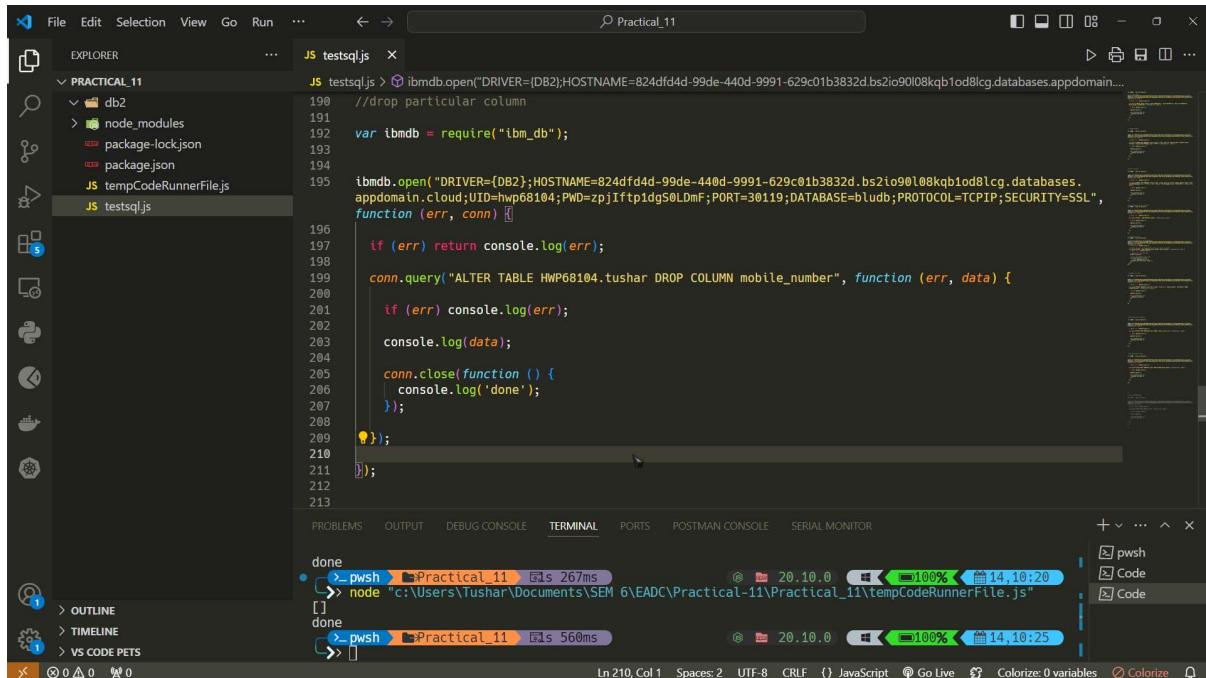
SQL

Export to CSV

F_NAME	L_NAME	EMAIL_ID	MOBILE_NUMBER
James	Bond	jamesbond007@gmail.com	777777
Tom	Cruise	tomcruise@gmail.com	1234567890

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

I used a script to drop the 'mobile\_number' column from the 'tushar' table in IBM Db2. The column deletion operation was successful, confirming the removal of the specified column from the database table.

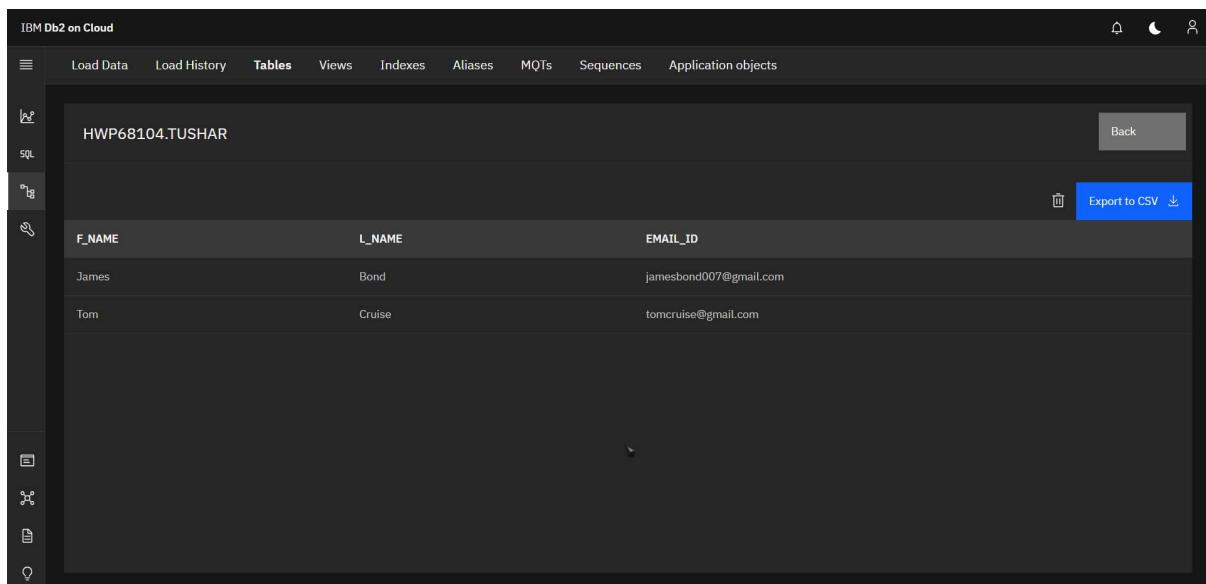


```

File Edit Selection View Go Run ... ← → Practical_11
EXPLORER JS testsql.js ...
PRACTICAL_11
db2
node_modules
package-lock.json
package.json
tempCodeRunnerFile.js
JS testsql.js
190 //drop particular column
191
192 var ibmdb = require("ibm_db");
193
194 ibmdb.open("DRIVER={DB2};HOSTNAME=824df4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=hvp68104;PWD=zpjIftp1dg50LdmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL", function (err, conn) {
195
196     if (err) return console.log(err);
197
198     conn.query("ALTER TABLE HWP68104.tushar DROP COLUMN mobile_number", function (err, data) {
199
200         if (err) console.log(err);
201
202         console.log(data);
203
204         conn.close(function () {
205             console.log('done');
206         });
207
208     });
209 }
210
211
212
213
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE PORTS POSTMAN CONSOLE SERIAL MONITOR
done
20.10.0 100% 14,10:20
node "c:\Users\tushar\Documents\SEM 6\EADC\Practical_11\tempCodeRunnerFile.js"
[]
done
20.10.0 100% 14,10:25
Ln 210, Col 1 Spaces:2 UTF-8 CRLF {} JavaScript Go Live Colorize: 0 variables Colorize

```

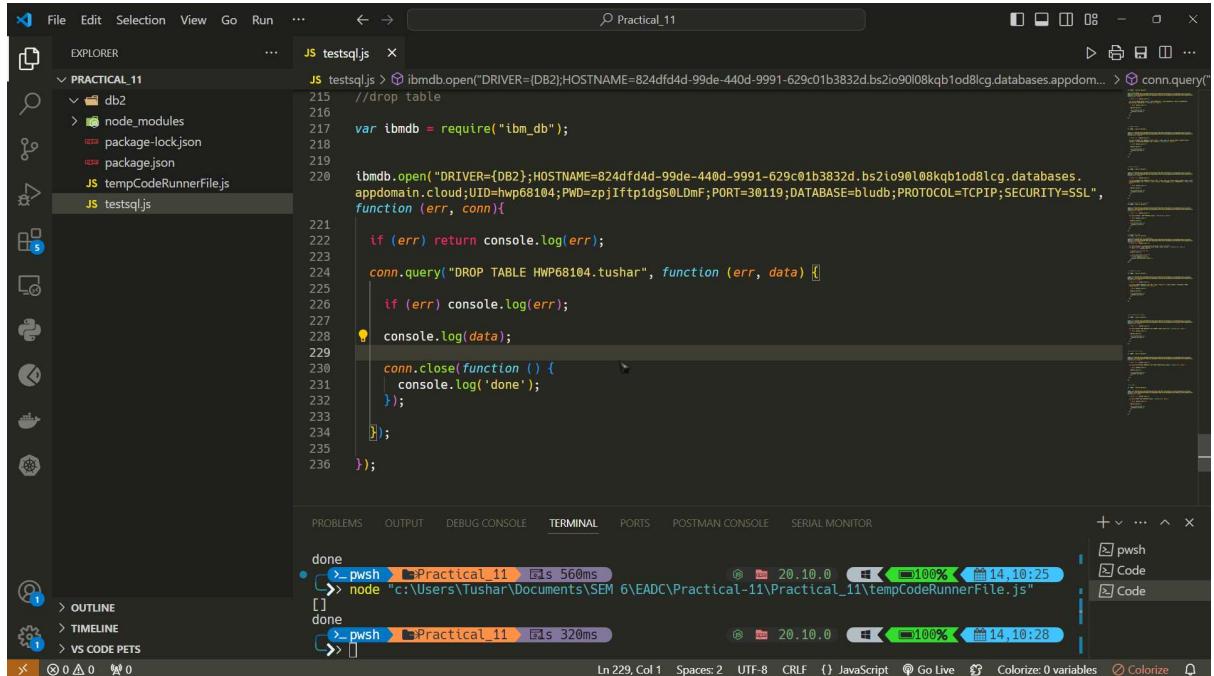
We can see below mobile\_number column deleted successfully:



F_NAME	L_NAME	EMAIL_ID
James	Bond	jamesbond007@gmail.com
Tom	Cruise	tomcruise@gmail.com

## » Practical 11.8 : Delete an entire table.

The 'tushar' table was successfully dropped from the database in IBM Db2.



```

File Edit Selection View Go Run ...
EXPLORER JS testsq1.js ...
PRACTICAL_11
db2
node_modules
package-lock.json
package.json
tempCodeRunnerFile.js
testsq1.js

JS testsq1.js > ibmdb.open("DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdom... > conn.query("
215 //drop table
216
217 var ibmdb = require("ibm_db");
218
219
220 ibmdb.open("DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdom...,
appdomain.cloud;UID=hwp68104;PWD=zpjIftp1dgS0LdmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL",
function (err, conn){
221
222 if (err) return console.log(err);
223
224 conn.query("DROP TABLE HWP68104.tushar", function (err, data) {
225
226 if (err) console.log(err);
227
228 console.log(data);
229
230 conn.close(function () {
231 console.log("done");
232 });
233
234 });
235
236 });

```

TERMINAL

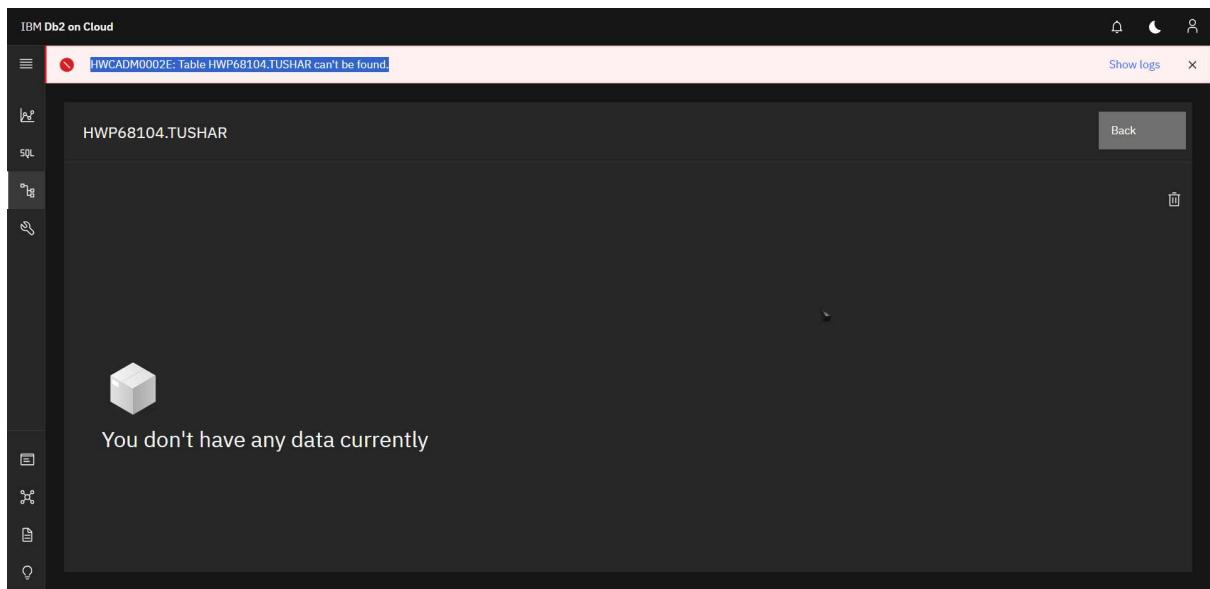
```

done
> psh > Practical_11 1s 560ms 20.10.0 100% 14,10:25
[ ]
done
> psh > Practical_11 1s 320ms 20.10.0 100% 14,10:28
[ ]

```

Ln 229, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript ⚡ Go Live ⚡ Colorize: 0 variables ⚡ Colorize

Here below you can see that TUSHAR table deleted successfully:



HWCADM002E: Table HWP68104.TUSHAR can't be found.

HWP68104.TUSHAR

You don't have any data currently

## » **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).**

➡ I successfully created a 'Employee' table. This table includes columns for EmployeeID, FirstName, LastName, Department and Salary ready for storing employee data.

### ✓ **Source Code:**

```
//TASK: ⏺ Database Setup:
var ibmdb = require("ibm_db");

var connString = "DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp68104;PWD=zpjIftp1dgS0LDmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL";

ibmdb.open(connString, function (err, conn) {
  if (err) return console.log(err);

  var createTableQuery = "CREATE TABLE Employees (EmployeeID INTEGER NOT NULL PRIMARY KEY, FirstName VARCHAR(255), LastName VARCHAR(255), Department VARCHAR(255), Salary DECIMAL(10, 2))";

  conn.query(createTableQuery, function (err, data) {
    if (err) console.log(err);

    console.log("Employees table created successfully.");

    conn.close(function () {
      console.log('Connection closed.');
    });
  });
});
```

➡ We can see that the Employee table has been successfully created in the Db2 UI.

The screenshot shows the IBM Db2 on Cloud web interface. On the left, there's a sidebar with icons for Load Data, Load History, Tables (which is selected), Views, Indexes, Aliases, MQTs, Sequences, and Application objects. Below this is a search bar labeled 'Find schemas or tables'. The main area has tabs for Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. Under the Tables tab, there's a 'Tables' section with a table header for Name, Schema, and Properties. A single row is listed: 'EMPLOYEES' under Name and 'HWP68104' under Schema. To the right, there's a 'Table definition' section for the 'EMPLOYEES' table. It shows the following columns:

Name	Data type	Nullable	Length	Scale
EMPLOYEEID	INTEGER	N	0	0
FIRSTNAME	VARCHAR	Y	255	0
LASTNAME	VARCHAR	Y	255	0
DEPARTMENT	VARCHAR	Y	255	0
SALARY	DECIMAL	Y	10	2

At the bottom of the table definition section, there's a 'View data' button. The status bar at the bottom left says 'Total: 1, selected: 0'.

## » TASK : 📁 Data Operations:

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

- ➡ I successfully inserted multiple records into the Employee table in Db2.

```
//TASK: 📁 Data Operations:
const ibmdb = require("ibm_db");

const connString = "DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp68104;PWD=zpjIftp1dgS0LDmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL";

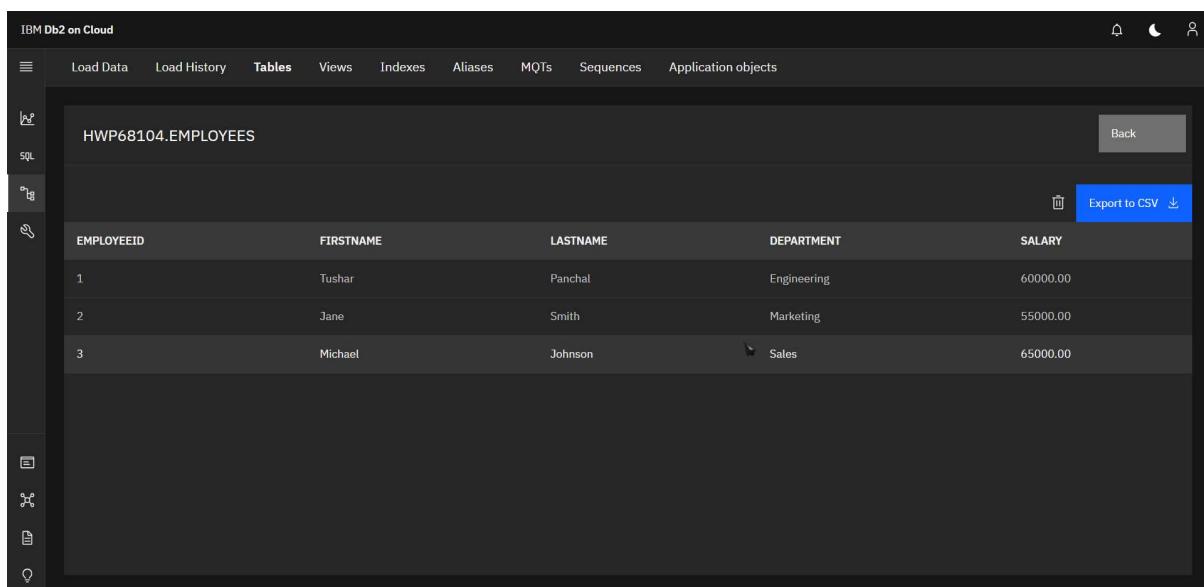
ibmdb.open(connString, function (err, conn) {
  if (err) return console.log(err);

  const insertQuery = "INSERT INTO Employees (EmployeeID, FirstName, LastName, Department, Salary) VALUES (?, ?, ?, ?, ?)";
  const employees = [
    [1, 'Tushar', 'Panchal', 'Engineering', 60000],
    [2, 'Jane', 'Smith', 'Marketing', 55000],
    [3, 'Michael', 'Johnson', 'Sales', 65000]
  ];

  employees.forEach(employee => {
    conn.query(insertQuery, employee, function (err) {
      if (err) console.log(err);
      else console.log("Record inserted successfully.");
    });
  });

  conn.close(() => console.log('Connection closed.'));
});
```

- ➡ Here as you can see in DB2 UI that data inserted successfully on table.



The screenshot shows the IBM Db2 on Cloud interface with the 'Tables' tab selected. The table 'HWP68104.EMPLOYEES' is displayed with the following data:

EMPLOYEEID	FIRSTNAME	LASTNAME	DEPARTMENT	SALARY
1	Tushar	Panchal	Engineering	60000.00
2	Jane	Smith	Marketing	55000.00
3	Michael	Johnson	Sales	65000.00

## 👉 **Retrieve all records from the Employee Table.**

- ➡ I have successfully retrieved all records into the Employee table in Db2.

### ✓ **Source Code:**

```
// Retrieve all records from the Employee Table.
const ibmdb = require("ibm_db");

const connString = "DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp68104;PWD=zpjIftp1dgS0LDmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL";

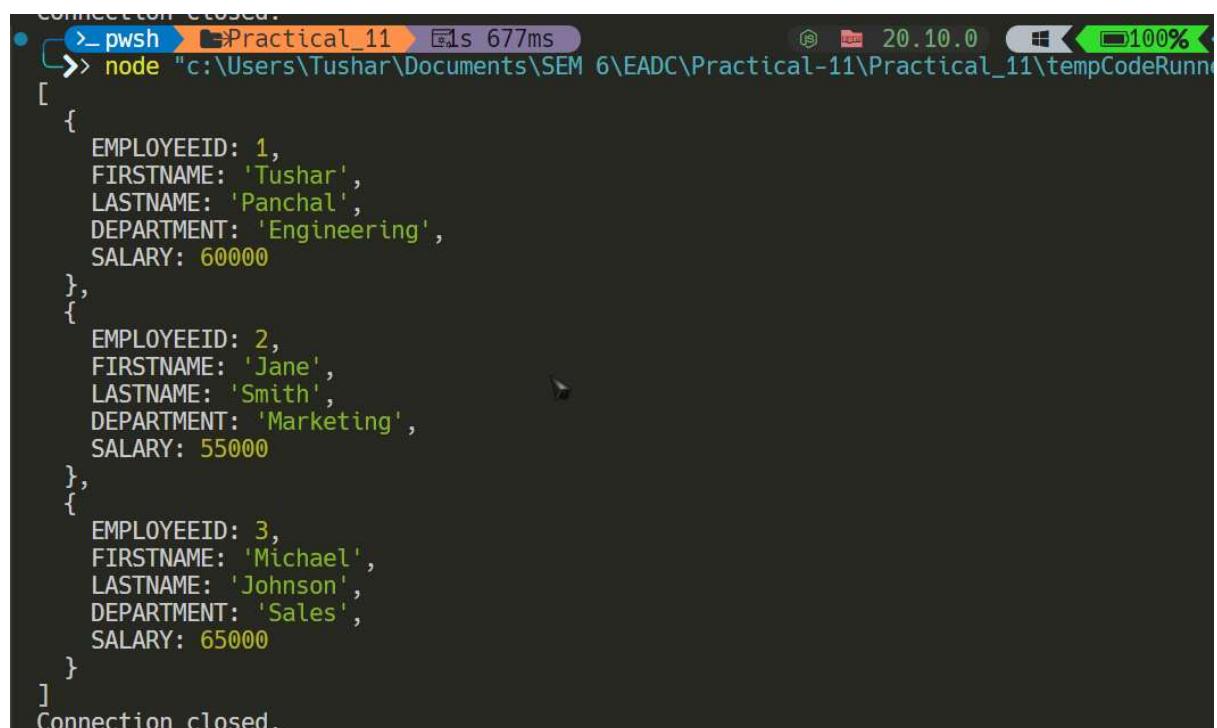
ibmdb.open(connString, function (err, conn) {
  if (err) return console.log(err);

  const selectQuery = "SELECT * FROM Employees";

  conn.query(selectQuery, function (err, data) {
    if (err) console.log(err);
    else console.log(data); // Output retrieved records

    conn.close(() => console.log('Connection closed.'));
  });
});
```

### ✓ **Output :**



The screenshot shows a terminal window with the following output:

```
connection closed.
[ { EMPLOYEEID: 1,
  FIRSTNAME: 'Tushar',
  LASTNAME: 'Panchal',
  DEPARTMENT: 'Engineering',
  SALARY: 60000 },
  { EMPLOYEEID: 2,
  FIRSTNAME: 'Jane',
  LASTNAME: 'Smith',
  DEPARTMENT: 'Marketing',
  SALARY: 55000 },
  { EMPLOYEEID: 3,
  FIRSTNAME: 'Michael',
  LASTNAME: 'Johnson',
  DEPARTMENT: 'Sales',
  SALARY: 65000 } ]
Connection closed.
```

We can see all employee details in console as above.

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

- ➡ I have successfully update record into the Employee table in Db2.

✓ **Source Code:**

```
//      Update the salary of an employee with EmployeeID 1 to $70,000.
const ibmdb = require("ibm_db");

const connString = "DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-
629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp68104;PWD=zpjIftp1dgS0LDmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL";
;

ibmdb.open(connString, function (err, conn) {
  if (err) return console.log(err);

  const updateQuery = "UPDATE Employees SET Salary = 70000 WHERE
EmployeeID = 1";
  const employeeID = 1;
  const newSalary = 70000;

  conn.query(updateQuery, [newSalary, employeeID], function (err, data) {
    if (err) console.log(err);
    else console.log("Salary updated successfully.");

    conn.close(() => console.log('Connection closed.'));
  });
});
```

- ➡ Here as you can see in DB2 UI that salary of an employee with EmployeeID 1 to \$70,000 is updated successfully on table.

EMPLOYEEID	FIRSTNAME	LASTNAME	DEPARTMENT	SALARY
1	Tushar	Panchal	Engineering	70000.00
2	Jane	Smith	Marketing	55000.00
3	Michael	Johnson	Sales	65000.00

## 👉 **Delete the record of the employee with EmployeeID 3.**

➡ I have successfully delete record into the Employee table in Db2.

### ✓ **Source Code:**

```
// Delete the record of the employee with EmployeeID 3.
const ibmdb = require("ibm_db");

const connString = "DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-
629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp68104;PWD=z
pjIftp1dgS0LDmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL";

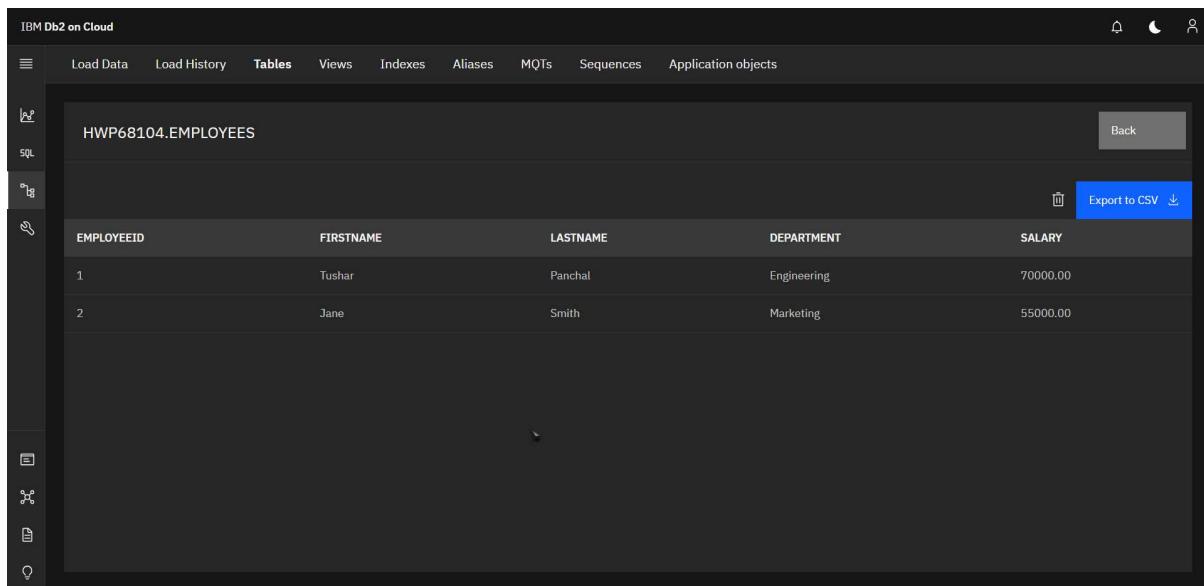
ibmdb.open(connString, function (err, conn) {
  if (err) return console.log(err);

  const deleteQuery = "DELETE FROM Employees WHERE EmployeeID = 3";
  const employeeIDToDelete = 3;

  conn.query(deleteQuery, [employeeIDToDelete], function (err, data) {
    if (err) console.log(err);
    else console.log("Record deleted successfully.");

    conn.close(() => console.log('Connection closed.'));
  });
});
```

➡ Here as you can see in DB2 UI that record of the employee with EmployeeID 3 from the Employees table is deleted successfully.



The screenshot shows the IBM Db2 on Cloud interface. The top navigation bar includes 'IBM Db2 on Cloud', 'Load Data', 'Load History', 'Tables' (which is selected), 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. On the left, there's a sidebar with icons for 'SQL', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The main area displays the 'HWP68104.EMPLOYEES' table. The table has columns: EMPLOYEEID, FIRSTNAME, LASTNAME, DEPARTMENT, and SALARY. There are two rows of data: one for employee ID 1 (Tushar, Panchal, Engineering, 70000.00) and one for employee ID 2 (Jane, Smith, Marketing, 55000.00). A 'Back' button is at the top right, and an 'Export to CSV' button is at the bottom right.

EMPLOYEEID	FIRSTNAME	LASTNAME	DEPARTMENT	SALARY
1	Tushar	Panchal	Engineering	70000.00
2	Jane	Smith	Marketing	55000.00

## » **TASK : 📁 Cleanup:**

### **Drop the Employees table from the database.**

- ➡ The Employee table was successfully dropped from the database in IBM Db2.

#### ✓ **Source Code:**

```
//TASK: 📁 Cleanup:  
//►Drop the Employees table from the database.  
const ibmdb = require("ibm_db");  
  
const connString = "DRIVER={DB2};HOSTNAME=824dfd4d-99de-440d-9991-  
629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;UID=hwp68104;PWD=z  
pjIftp1dgS0LDmF;PORT=30119;DATABASE=bludb;PROTOCOL=TCPIP;SECURITY=SSL";  
  
ibmdb.open(connString, function (err, conn) {  
  if (err) return console.log(err);  
  
  const dropTableQuery = "DROP TABLE Employees";  
  
  conn.query(dropTableQuery, function (err, data) {  
    if (err) console.log(err);  
    else console.log("Employees table dropped successfully.");  
  
    conn.close(() => console.log('Connection closed.'));  
  });  
});
```

- ➡ Here as you can see in DB2 UI that table named employee is dropped successfully.

The screenshot shows the IBM Db2 on Cloud web interface. The main page title is 'HWP68104.EMPLOYEES'. On the left, there's a sidebar with icons for Tables, SQL, and other database management options. The main content area displays a table structure with one column labeled 'EMPLOYEE\_ID'. A prominent red error message at the top left states 'HWCADM0002E: Table HWP68104.EMPLOYEES can't be found.' Below the table, a message says 'You don't have any data currently'. The overall theme is dark, and the error message is highlighted in red.