

Assignment -3

Assignment Date	21 September 2022
Student Name	P.Ramprakash
Student Roll Number	420619104029
Maximum Marks	2 Marks
Project	Plasma Donor Application

Question-1:

1) Create user table with user with email, user name, roll number, password.

Solution:

```
CREATE TABLE user(  
    email VARCHAR(50) ,  
    username VARCHAR(50) ,  
    roll_number VARCHAR(50) ,  
    password VARCHAR(50)  
);
```

```
INSERT INTO user (email,username,roll_number,password)  
VALUES ('jkrkumar1801@gmail.com','jayakumar','420619104019','jkjk1801');
```

```
INSERT INTO user (email,username,roll_number,password)  
VALUES ('ramp894028@gmail.com','ramprakash','420619104029','rp894028');
```

```
INSERT INTO user (email,username,roll_number,password)  
VALUES ('sivasiva6735@gmail.com','balaji','420619104010','balaji6735');
```

```
INSERT INTO user (email,username,roll_number,password)  
VALUES ('akumarancse843@gmail.com','arasakumaran','420619104005','akumar8');
```

The screenshot shows the IBM Db2 on Cloud web interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected, and the table 'VBS30730.USER' is displayed. The table has four columns: EMAIL, USERNAME, ROLL_NUMBER, and PASSWORD. The data is as follows:

EMAIL	USERNAME	ROLL_NUMBER	PASSWORD
akumarancse843@gmail.com	arasakumaran	420619104005	akumar8
jkrkumar1801@gmail.com	jayakumar	420619104019	jkjk1801
ramp894028@gmail.com	ramprakash	420619104029	rp894028
sivasiva6735@gmail.com	balaji	420619104010	balaji6735

Buttons for 'Back', 'Export to CSV', and 'Download' are visible in the top right corner of the table view.

Question-2:

2) Perform UPDATE, DELETE Queries with user table.

Solution:

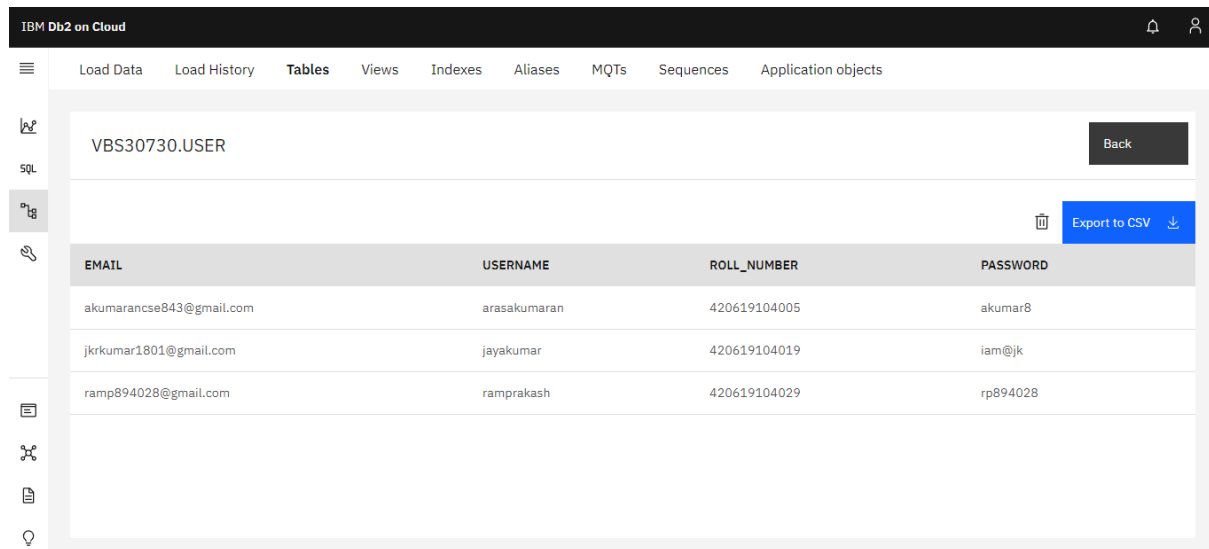
UPDATE user

SET password = 'iam@jk'

WHERE password = 'jkjk1801'

DELETE FROM user

WHERE username = 'balaji' ;



The screenshot shows the IBM Db2 on Cloud console interface. At the top, there's a navigation bar with 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected. Below the navigation bar, the table 'VBS30730.USER' is displayed. To the right of the table name is a 'Back' button. Below the table name, there's a table structure view showing columns: EMAIL, USERNAME, ROLL_NUMBER, and PASSWORD. The table contains three rows of data. To the right of the table structure view, there's a 'Export to CSV' button and a download icon.

EMAIL	USERNAME	ROLL_NUMBER	PASSWORD
akumarancse843@gmail.com	arasakumaran	420619104005	akumar8
jkrcumar1801@gmail.com	jayakumar	420619104019	iam@jk
ramp894028@gmail.com	ramprakash	420619104029	rp894028

Question-3:

3) Connect python code to db2.

Solution:

import ibm_db

```
hostname="824dfd4d-99de-440d-9991629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud"
uid="vbs30730"
pwd="JjuONYwOmUXnbHNx"
driver="{IBM DB2 ODBC DRIVER}"
db="bludb"
port="30119"
protocol="TCPIP"
cert="certificate.crt"
```

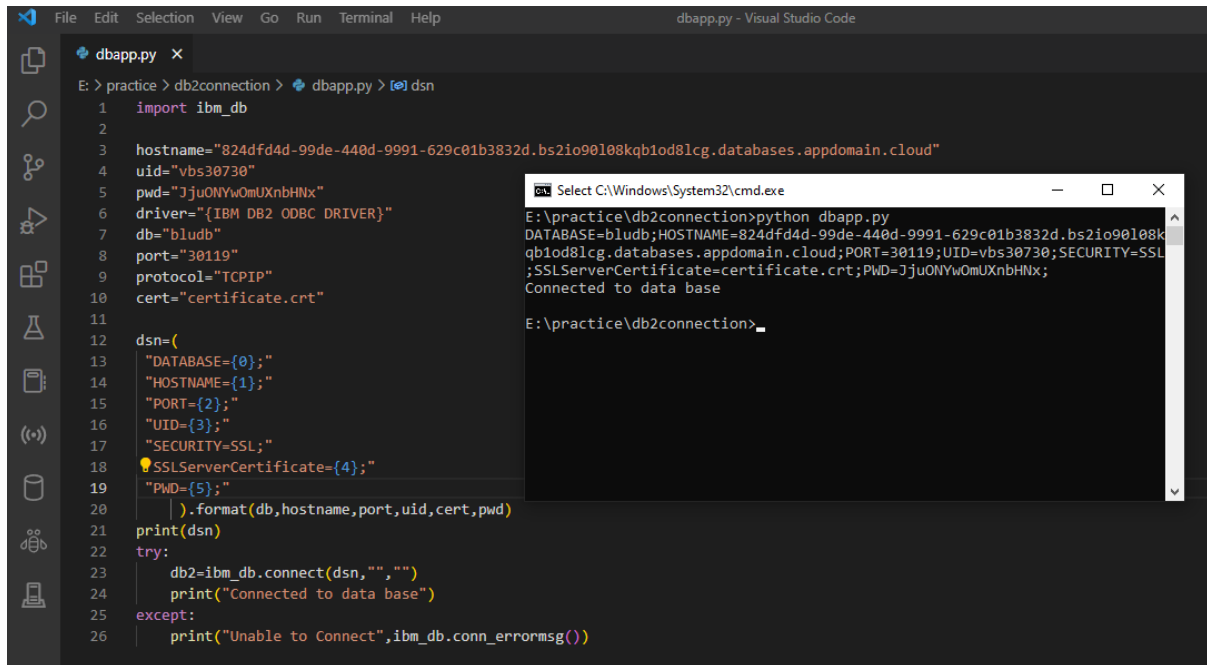
```
dsn=(
    "DATABASE={0};"
    "HOSTNAME={1};"
    "PORT={2};"
    "UID={3};"
    "SECURITY=SSL;"
    "SSLServerCertificate={4};"
    "PWD={5};"
).format(db,hostname,port,uid,cert,pwd)
print(dsn)
```

try:

```
db2=ibm_db.connect(dsn,"","")  
print("Connected to data base")
```

except:

```
print("Unable to Connect",ibm_db.conn_errormsg())
```



The screenshot shows a Visual Studio Code editor with a file named `dbapp.py` open. The script is a Python program that attempts to connect to an IBM DB2 database using the `ibm_db` module. It defines a DSN (Data Source Name) with various parameters like `DATABASE`, `HOSTNAME`, `PORT`, `UID`, `SECURITY`, `SSLServerCertificate`, and `PWD`. The script then uses `ibm_db.connect(dsn, "", "")` to establish the connection. If successful, it prints "Connected to data base". If an exception occurs, it prints "Unable to Connect" along with the error message from `ibm_db.conn_errormsg()`.

The terminal window shows the execution of the script. It displays the DSN parameters and the successful connection message: "Connected to data base".

Question-4:

4) Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password. If the user is valid shown the welcome page.

Solution:

from flask import *

```
app = Flask(__name__)
```

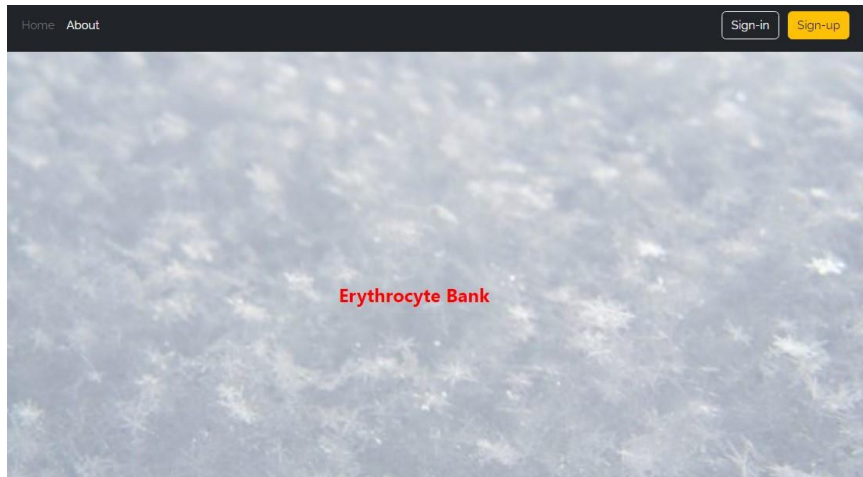
```
@app.route('/')  
def hello_world():  
    return render_template("index.html")
```

```
@app.route('/signup')  
def signup():  
    return render_template('signup.html')
```

```
@app.route('/signin')  
def signin():  
    return render_template('signin.html')
```

```
@app.route('/about')  
def about():  
    return render_template('about.html')
```

```
if __name__ == '__main__':  
    app.run(debug=True)
```



Sign in

Email address

Password

LOGIN

[Forgot password?](#)



Register

Create your account

First Name

Last Name

Email

Password

Confirm Password

☐ I accept the [Terms of Use & Privacy Policy](#)

Register Now

