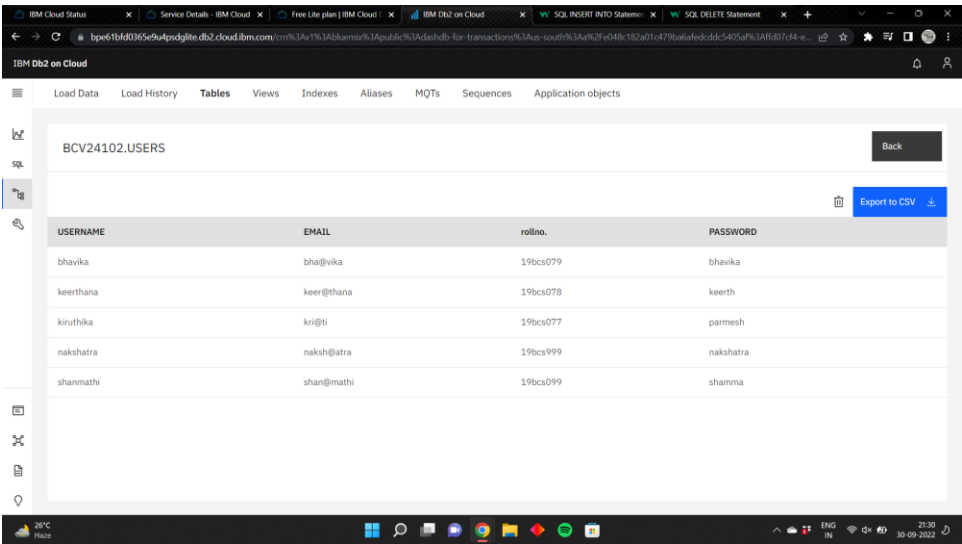


Assignment 2

Bhavika Arage

1. Create User table with user with email.username.roll number password.

Original table

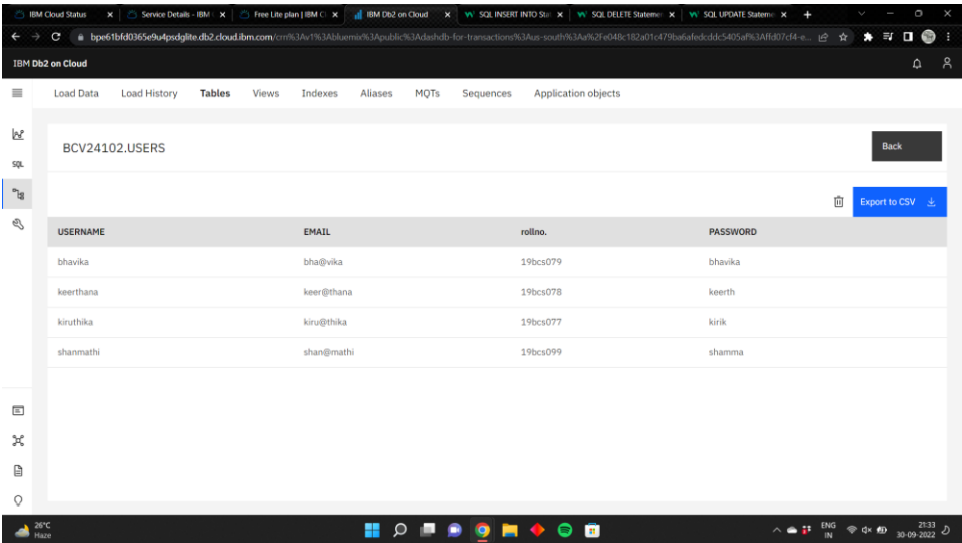


The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, displaying the table 'BCV24102.USERS'. The table structure is as follows:

USERNAME	EMAIL	rollno.	PASSWORD
bhavika	bha@vika	19bcs079	bhavika
keerthana	keer@thana	19bcs078	keerth
kiruthika	kri@ti	19bcs077	parmesh
nakshatra	naksh@atra	19bcs999	nakshatra
shanmathi	shan@mathi	19bcs099	shamma

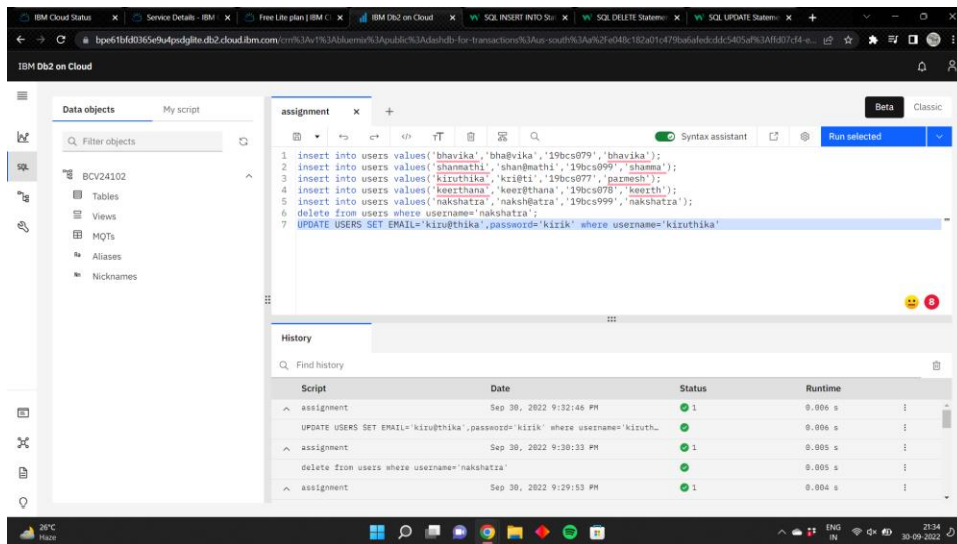
2. Perform UPDATE DELETE Queries with user table

After update and delete



The screenshot shows the IBM Db2 on Cloud web interface after performing updates. The table 'BCV24102.USERS' now contains the following data:

USERNAME	EMAIL	rollno.	PASSWORD
bhavika	bha@vika	19bcs079	bhavika
keerthana	keer@thana	19bcs078	keerth
kiruthika	kiru@thika	19bcs077	kirik
shanmathi	shan@mathi	19bcs099	shamma



SQL:

```
insert into users values('bhavika','bha@vika','19bcs079','bhavika');
insert into users values('shanmathi','shan@mathi','19bcs099','shamma');
insert into users values('kiruthika','kri@ti','19bcs077','pamesh');
insert into users values('keerthana','keer@thana','19bcs078','keerth');
insert into users values('nakshatra','naksh@atra','19bcs999','nakshatra');
delete from users where username='nakshatra';
UPDATE USERS SET EMAIL='kiru@thika',password='kirik' where username='kiruthika'
```

3. Connect python code to db2.

```
conn = ibm_db.connect("DATABASE=bludb;"
                      "HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;"
                      "PORT=30376;"
                      "SECURITY=SSL;"
                      "SSLServerCertificate=C:/Users/bhaua/OneDrive/Desktop/SEM Stuff KCT/IBM/DigiCertGlobalRootCA.crt;"
                      "UID=bcv24102;"
                      "PWD=ilZgx2Zokf2nJOCW;", "", "")
```

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 show the welcome page

App.py

```

from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re

app = Flask(__name__)
app.secret_key = 'Zenik'

conn = ibm_db.connect("DATABASE=bludb;"
                      "HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;"
                      "PORT=30376;"
                      "SECURITY=SSL;"
                      "SSLServerCertificate=C:/Users/bhaua/OneDrive/Desktop/SEM Stuff KCT/IBM/DigiCertGlobalRootCA.crt;"
                      "UID=bcv24102;"
                      "PWD=ilZgx2Zokf2nJOCW;", "", "")

@app.route('/')
@app.route('/home')
def home():
    return render_template('home.html', title='Home', msg=" ")

@app.route('/dashboard')
def dashboard():
    sql = "SELECT * FROM USERS WHERE username =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, session['username'])
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    return render_template('dashboard.html', title='Dashboard', account=account)

@app.route('/logout')
def logout():
    session.pop('Loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect('/')

@app.route('/login', methods=['GET', 'POST'])
def login():
    global userid
    msg = " "

    if request.method == "POST":
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM USERS WHERE username =? AND password =?"

```

```

stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt, 1, username)
ibm_db.bind_param(stmt, 2, password)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)
if account:
    session['Loggedin'] = True
    session['id'] = account['USERNAME']
    userid = account['USERNAME']
    session['username'] = account['USERNAME']
    return redirect('/dashboard')
else:
    msg = "Incorrect login credentials"

return render_template('login.html', title='Login', msg=msg)

```

```

@app.route('/register', methods=['GET', 'POST'])
def register():
    msg = " "
    if request.method == "POST":
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        password1 = request.form['password1']
        rollno = request.form['rollno']
        sql = "SELECT * FROM USERS WHERE username =? or email=? "
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = "Account already exists"
        elif password1 != password:
            msg = "re-entered password doesnt match"
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = "Username should be only alphabets and numbers"
        else:
            sql = "INSERT INTO USERS VALUES (?, ?, ?, ?)"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email)
            ibm_db.bind_param(stmt, 3, rollno)
            ibm_db.bind_param(stmt, 4, password)
            ibm_db.execute(stmt)
            return redirect('/login')
    return render_template('register.html', msg=msg, title="Register")

```

```

if __name__ == '__main__':

```

```
app.run()
```

Respective html pages were created. (layout,home,login,registration,dashboard)

