

ASSIGNMENT 2

Student Name	Ms.Aarthi Suresh Kumar
Student Roll Number	195001003

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

INSERTION:

The screenshot shows the IBM Db2 on Cloud console. On the left, the 'Data objects' tab is active, displaying a search bar and a list of objects including 'WYM47343'. The main area shows a script editor with the following SQL code:

```
1 insert into user values('u1@gmail.com','u1','01','u101');
2 insert into user values('u2@gmail.com','u2','02','u202');
3
```

Below the script editor, the 'History' tab is active, showing a table of executed scripts:

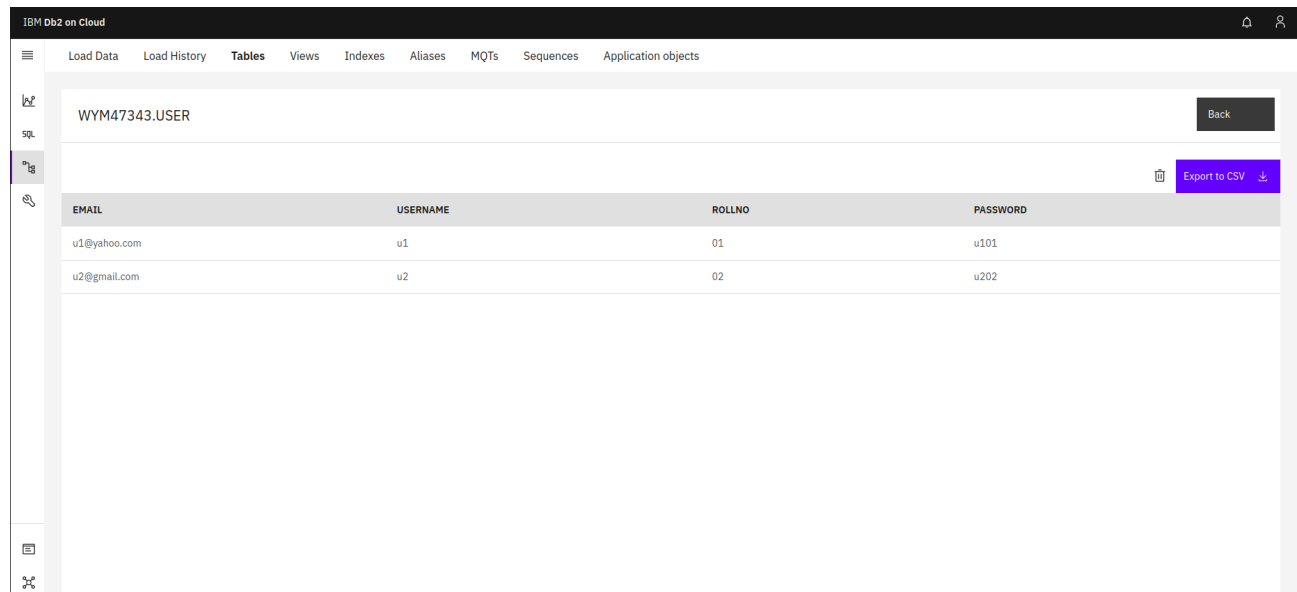
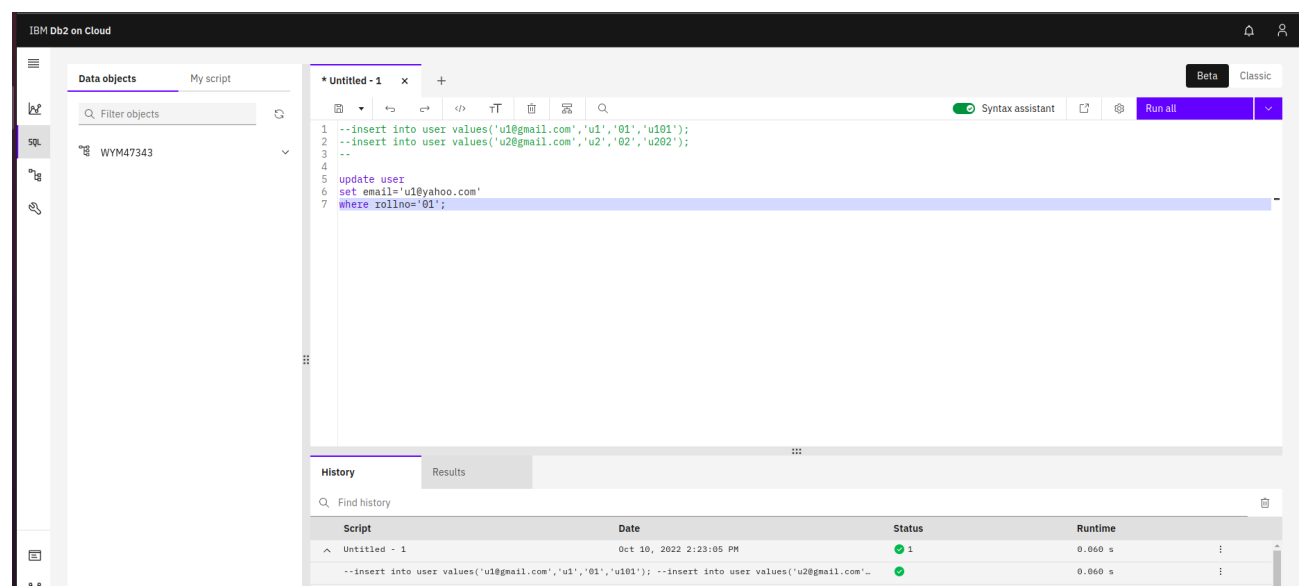
Script	Date	Status	Runtime
Untitled - 1	Oct 10, 2022 2:19:44 PM	2	0.040 s
insert into user values('u1@gmail.com','u1','01','u101')		✓	0.020 s
insert into user values('u2@gmail.com','u2','02','u202')		✓	0.020 s

The screenshot shows the IBM Db2 on Cloud console with the 'USER' table selected. The table structure is defined as follows:

EMAIL	USERNAME	ROLLNO	PASSWORD
u1@gmail.com	u1	01	u101
u2@gmail.com	u2	02	u202

2. Perform UPDATE,DELETE Queries with user table

UPDATION:



DELETION:

IBM Db2 on Cloud

WYM47343

Filter objects

Untitled - 1

Syntax assistant

Run all

```
1 --insert into user values('u1@gmail.com','u1','01','u101');
2 --insert into user values('u2@gmail.com','u2','02','u202');
3 --
4
5 --update user
6 --set email='u1@yahoo.com'
7 --where rollno='01';
8
9
10 delete from user
11 where username='u1';
```

History

Results

Find history

Script	Date	Status	Runtime
^ Untitled - 1	Oct 10, 2022 2:26:46 PM	1	0.005 s
--insert into user values('u1@gmail.com','u1','01','u101'); --insert into user values('u2@gmail.com'...			0.005 s

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

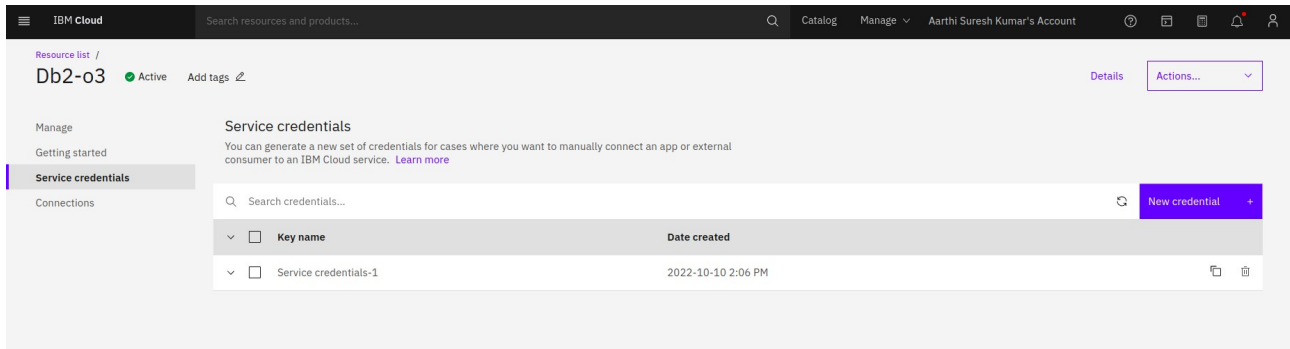
WYM47343.USER

Back

Export to CSV

EMAIL	USERNAME	ROLLNO	PASSWORD
u2@gmail.com	u2	02	u202

3. Connect python code to db2.



```
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;Security=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=wym47343;PWD=RfUrGhT3TSiZ6ZWY;","","")
```

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
server.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
```

Created on Mon Oct 10 14:37:25 2022

```
@author: aarthi
"""
```

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

```
app=Flask(__name__, template_folder='templates')
app.secret_key='a'
```

```
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;Security=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=wym47343;PWD=RfUrGhT3TSiZ6ZWY;","","")
```

```
@app.route('/', methods=['GET','POST'])
def register():
    msg=""
```

```
    if request.method=='POST':
```

```
username=request.form['username']
password=request.form['password']
email=request.form['email']
rollno=request.form['rollno']
sql="INSERT INTO user VALUES(?,?,?,?)"
```

```
stmt=ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt,1,email)
ibm_db.bind_param(stmt,2,username)
ibm_db.bind_param(stmt,3,rollno)
ibm_db.bind_param(stmt,4,password)
ibm_db.execute(stmt);
```

```
msg='Succesfully registered!'
```

```
return redirect(url_for('login'))
```

```
elif request.method=='GET':
    msg='Please fill out the form'
    return render_template('register.html',msg=msg)
```

```
@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 user 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"]
            msg='Logged in successfully'
```

```
            return render_template("welcome.html",msg=msg,username=userid)
        else:
            msg='Incorrect username/password'
            return render_template('login.html', msg=msg)
```

```
elif request.method=='GET':  
    return render_template('login.html')
```

```
if __name__=='__main__':  
    app.run(host='0.0.0.0')
```

register.html

```
<html>  
<body>  
    <h6 style="color:red">{{ msg }}</h6>  
    <form action="/" method="POST">  
        Username* <input type="text" name="username" placeholder="Enter username "><br/><br/>  
        Password* <input type="password" name="password"><br/><br/>  
        Email* <input type="email" name="email" placeholder="Enter email"><br/><br/>  
        Roll number*<input type="number" name="rollno" placeholder="Enter roll no."><br/><br/>  
        <input type="submit" value="Register">  
    </form>  
</body>  
</html>
```

login.html

```
<html>  
<body>  
    <h5 style="color:red">{{ msg }}</h5>  
    <form action="/login" method="POST">  
        Username <input type="text" name="username" placeholder="Enter username"><br/><br/>  
        Password <input type="password" name="password" placeholder="*****"><br/><br/>  
        <input type="submit" value="Login">  
    </form>  
</body>  
</html>
```

welcome.py

```
<h1>Welcome {{username}}!</h1>
```

```
<h6>{{ msg }}</h6>
```

REGISTRATION PAGE:

Please fill out the form

Username*

Password*

Email*

Roll number*

LOGIN PAGE:
redirected to login page after successful registration

Username

Password

Login

WELCOME PAGE:
redirected to welcome page on successful login

Welcome u4 !

Logged in successfully

Redirected back to login page in the case of incorrect credentials

Incorrect username/password

Username

Password

Login