

Assignment -2

Team Id	PNT2022TMID29528
Student Name	T.PREETHA
Roll_Number	510419106020

1.Create user table with user email,username,password,roll no

2. perform update ,delete queries with user table

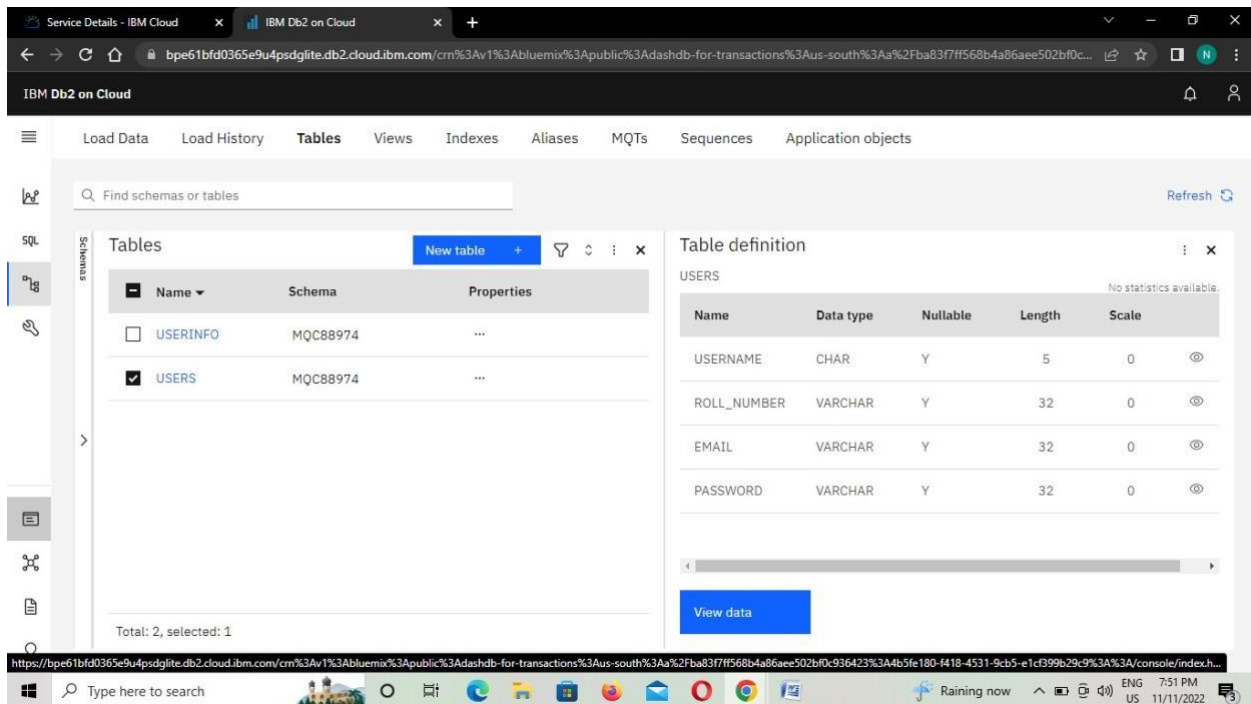
3.connect python code to database2

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 amnd password.if the user is valid show the welcome page

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

CREATE TABLE user (

username char,roll_number varchar(32),email
varchar(32), password varchar(32))



2. perform UPDATE , DELETE Queries with user table.

INSERT Statement:

```
insert into users values('niha',18,'gamma@gmail.com',72); insert into users
values('amir',12,'xy@gmail.com',78); insert into users
values('syed',14,'zeno@gmail.com',67); insert into users values('niva',25,
'beta@gmail.com',56); insert into users values('ashu',45,'alphagmail.com',52);
```

MQC88974.USERS

USERNAME	ROLL_NUMBER	EMAIL	PASSWORD
amir	12	xy@gmail.com	78
ashu	45	alphagmail.com	52
niha	18	gamma@gmail.com	77
niva	25	beta@gmail.com	56
syed	14	zeno@gmail.com	67

UPDATE Statement:

```
update users SET username='priya' WHERE
roll_number='12'
```

Service Details - IBM Cloud | IBM Db2 on Cloud | IBM Db2 on Cloud

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fba837f7ff568b4a86aee502bf0c...

IBM Db2 on Cloud

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

MQC88974.USERS

Back

Export to CSV

USERNAME	ROLL_NUMBER	EMAIL	PASSWORD
ashu	45	alphagmail.com	52
niha	18	gamma@gmail.com	77
niva	25	beta@gmail.com	56
priya	12	xy@gmail.com	78
syed	14	zeno@gmail.com	67

Type here to search

Rain... 8:24 PM 11/11/2022

DELETE Statemnet:

delete from users WHERE roll_number='18'

Service Details - IBM Cloud | IBM Db2 on Cloud | IBM Db2 on Cloud

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fba837f7ff568b4a86aee502bf0c...

IBM Db2 on Cloud

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

MQC88974.USERS

Back

Export to CSV

USERNAME	ROLL_NUMBER	EMAIL	PASSWORD
ashu	45	alphagmail.com	52
niva	25	beta@gmail.com	56
priya	12	xy@gmail.com	78
syed	14	zeno@gmail.com	67

Type here to search

Rain... 8:26 PM 11/11/2022

3. Connect python with db2

```
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud,;PORT=30875;security=SSL;SSLServercertificate==DigiCertGlobalRootCA.cr;UID=mqc88974;PWD=mbf14fY4F8mTnrXJ;")
```



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 f

```
from flask import Flask, render template, request, redirect, url for,  
session from flask mysqlldb import MySQL import  
MySQLldb.cursors import reapp = Flask( name ) app.secret key
```

```

= 'your secret key' app.config['MYSQL HOST'] = 'localhost'
app.config['MYSQL USER'] = 'root' app.config['MYSQL
PASSWORD'] = 'your password' app.config['MYSQL DB'] =
'geeklogin' mysql =
MySQL(app)
@app.route('/')
@app.route('/login', methods=['GET', 'POST'])
def login():
msg = 'if request.method == 'POST' and 'username' in request.form and
'password' in request.form:
username = request.form['username'] password =
request.form['password'] cursor =
mysql.connection.cursor(MySQLdb.cursors.DictCursor)
cursor.execute('SELECT * FROM accounts WHERE username = % s
AND password = % s', (username, password,
)) account = cursor.fetchone() if
account:
session['loggedin'] = True session['id']
= account['id']
session['username'] = account['username'] msg
= 'Logged in successfully !' return render
template('index.html', msg = msg)

```

else:

msg = 'Incorrect username / password !' return

render template('login.html', msg = msg)

@app.route('/logout') def logout():

session.pop('loggedin', None) session.pop('id',

None) session.pop('username', None) return

redirect(url for('login'))

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

register(): msg

= "

if request.method == 'POST' and 'username' in request.form and
'password' in request.form and 'email' in

request.form : username =

request.form['username'] password =

request.form['password'] email =

request.form['email'] cursor =

mysql.connection.cursor(MySQLdb.cursors.

DictCursor) cursor.execute('SELECT * FROM

accounts WHERE username = % s',

(username,)) account =

cursor.fetchone() if

account:

```

msg = 'Account already exists !' elif not
re.match(r'^@]+@[^@]+\.[^@]+', email): msg = 'Invalid
email address !' elif not re.match(r'[A-Za-z0-
'Username must contain only characters elif not
username or not password or not email:
msg = 'Please fill out the form !' else:
cursor.execute('INSERT INTO accounts VALUES (NULL, %
s, % s, % s)', (username, password, email, ))
mysql.connection.commit() msg = 'You
have successfully registered !' elif
request.method == 'POST':
msg = 'Please fill out the form !' return render
template('register.html', msg = msg)
9]+' , username):
and numbers !

```



