## **Assignment -2**

## **Python Programming**

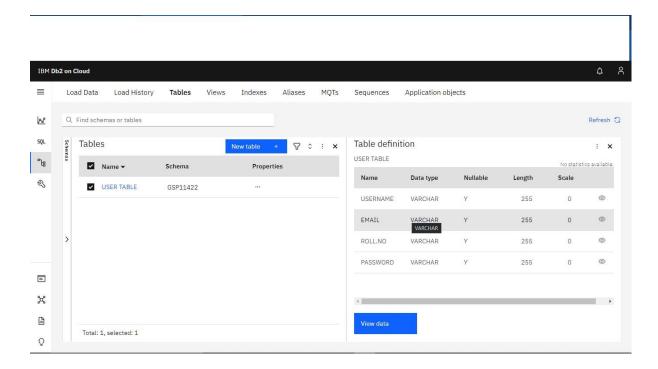
Assignment Date	28 Oct 2022
Student Name	Yogesh RB
Team id	PNT2022TMID31186
Maximum Marks	2 Marks

## **Question-1:**

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

#### **Solution:**

Create table students



## **Question-2:**

Perform UPDATE, DELETE queries with user table **Solution:** 

Insert into user values ("sathish",23, 'sarkarsathish@gmail.com', "s@thish");

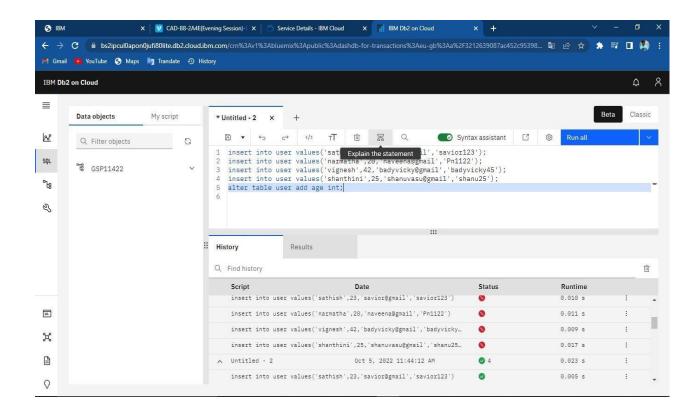
Insert into user values ("shanthini",25, 'shanu@gmail.com', "shanu@cse");

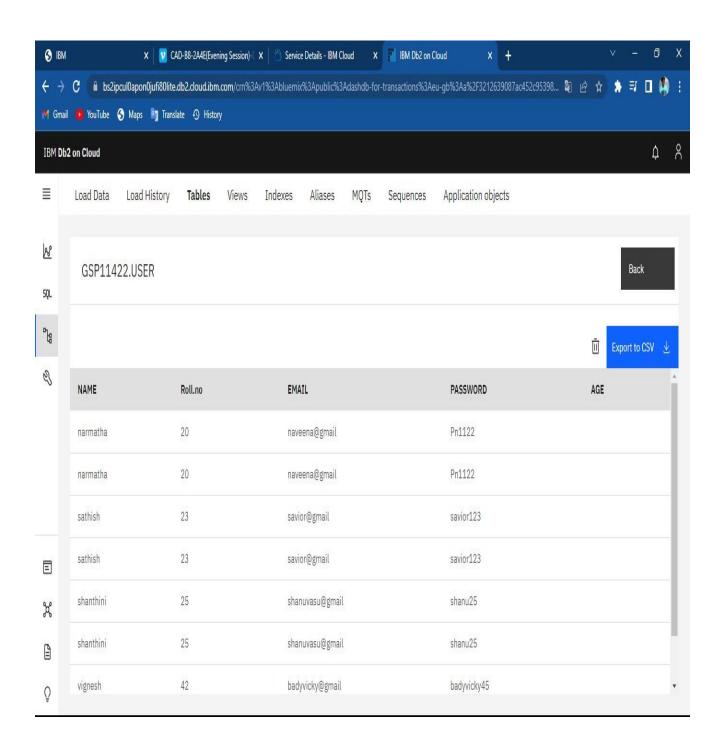
Insert into user values ("Narmatha",20, 'narmatha@gmail.com', "narmu");

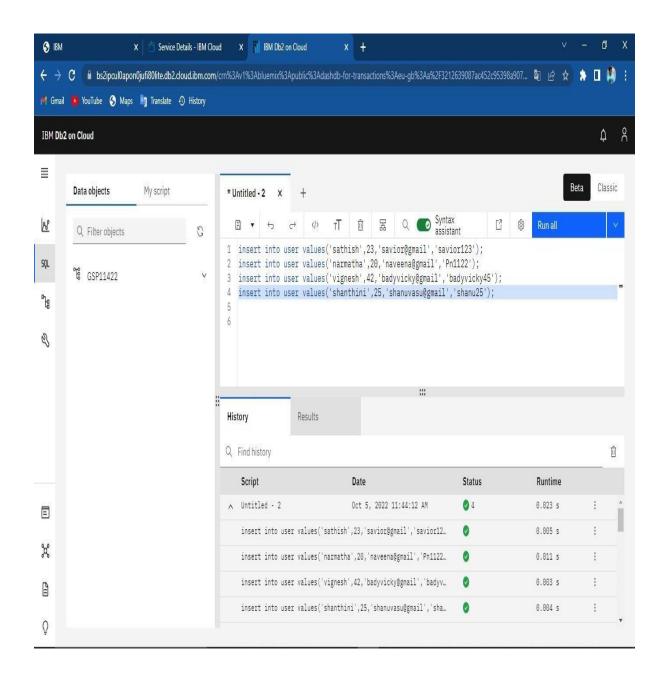
Insert into user values ("vignesh",37, 'vignesh@gmail.com',"vicky");

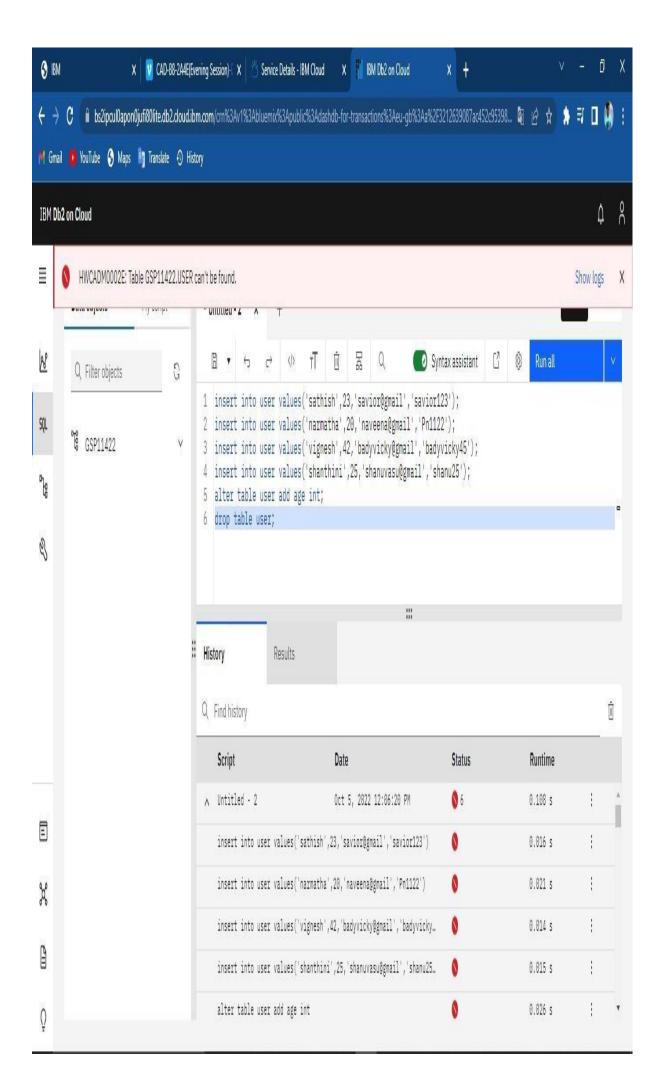
Alter table user add age int;

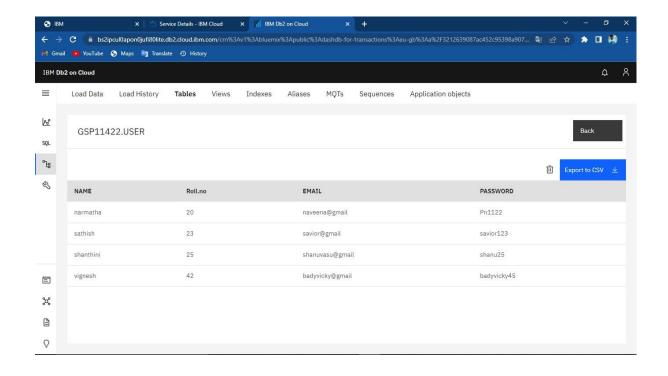
Drop table user;











# **Question-3:**

Connect python code to db2 **Solution:** from flask import Flask, render\_template, request, redirect, url\_for, session import ibm\_db import re

```
app = Flask(____name_)
app.scret_key = 'a' conn
=
ibm_db.conect("DATABASE=;HOSTNAME=;PORT=;SECURITY=SSL;SSL
ServerCertificate=;UID=;PhD=", ' ', ' ')

@app.route('/') def
home():
```

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

return render\_template('home.html')

```
def login(): global
   userid msg = ' '
if request.method == 'POST':
     username
                            request.form['username']
                                                         password
     request.form['password'] return render_template('home.html') sql =
     "SELECT * FROM Users WHERE userame=? 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['Loggeddin'] = True session['id'] =
     account['username']
                                   userid
     account['USERNAME'] session['username'] =
     acccount['USERNAME'] else:
        msg = 'Incorrect username/password' return
        render_template('login.html',
                                          msg=msg)
        @app.route('/register',
                                   methods=['GET',
        'POST']) def register():
   if
                                'POST':
        request.method
                                           username
     request.form['username'] email = request.form['email']
     password = request.form['password'] sql = "SELECT *
     FROM users
                      WHERE
                                username =?"
     ibm_db.prepare(conn, sql) ibm_db.bind_param(stmt,
     1.
          username)
                      ibm_db.execute(stmt)
                                              account
     ibm.db.fetch_assoc(stmt) print(account) if account:
        msg = "Accont already exists!"
     elif not re.match(r'[^{\circ}@]+@[^{\circ}@]+\\.[^{\circ}@]+', email):
        msg = "format does not match"
     elif not re.match(r'[A-Za-z0-9+', username):
        msg = "name must contain characters and numbers" else:
        insert_sql = "ISERT INTO users VALUES(?, ?, ?)"
                          ibm_db.prepare(conn.insert_sql)
        prep_stmt
```

```
ibm_db.bind_param(prep_stmt,
                                         1,
                                              username)
                                           2,
        ibm_db.bnd_param(prep_stmt,
                                                  email)
        ibm_db.bind_param(prep_stmt,
                                         3.
                                              password)
        ibm_db.execute(prep_stmt) msg = "You have
        successfully registered"
   elif (request.method == "POST"): msg == "Please
     fill out the form" return
     render_template('register.html', msg=msg)
      @app.route('/dashboard') def dash():
   return render_template('dashboard.html')
 @app.route('/apply', methods=['GET","POST'])
def app(): msg = ' ' if request.method ==
 "POST": username = request.form['username']
email = request.form['email'] qualification =
request.form['qualification']
                                 skills
                                            =
request.form['skills'] jobs = request.form['s']
                ibm_db.prepare(conn,
 stmt
ibm_db.bind_param(stmt,
                              1,
                                    username)
 ibm_db.execute(stmt)
                             account
ibm_db.fetch_assoc(stmt) print(account)
if account():
       msg = "there is only 1 job position" return
        render_template('apply.html', msg=msg)
     insert_sql = "INSERT INTO job VALUES(?, ?, ?, ?, ?)"
                       ibm_db.prepare(conn,
                                                insert_sql)
     prep_stmt
                                         1,
     ibm_db.bind_param(prep_stmt,
                                                username)
     ibm_db.bind_param(prep_stmt,
                                           2,
                                                    email)
```

ibm\_db.bind\_param(prep\_stmt,

3,

qualification)

```
ibm_db.bind_param(prep_stmt,
                                           4,
                                                     skills)
     ibm_db.bind_param(prep_stmt,
                                            5,
                                                      jobs)
     ibm_db.execute(prep_stmt)
                                   msg
                                          = "You
                                                      have
     successfully applie for job" session['Loggedin'] = True
     TEXT = "Hello user, a new application for job position" + job + isrequested
elif request.method == "POST" msg =
      "Please fill out the form"
     return render_template('register.html', msg=msg)
 @app.route('/display')
 def display():
   print session["username"], session['id'] cursor =
   mysql.connection.cursor() cursor.execute('SELECT*FROM job
   WHERE userid=%s', (sessio['id'],)) account = cursor.fetchone()
   print("accountdisplay", account
```

### **Question-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 **Solution:** 

from flask import Flask, render\_template, request, redirect, url\_for, session import ibm\_db import request

conn =

```
ibm_db.conect("DATABASE=bludb;HOSTNAME=21fecfd8
-47b7-4937-840d-
d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.c
loud;PORT=31864;SECURITY=SSL;SSLServerCertificate=
DigiCertGlobalRootCA.crt;USERNAME=gsp11422;PASSW
ORD=ixh307unZIzgLLB9", '', '')
@app.route('/') def
home():
  return render_template('home.html')
@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 userame=?
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['Loggeddin'] = True session['id'] =
       account['username']
                                  userid
       account['USERNAME'] session['username'] =
       account['USERNAME'] msg='Logged
       successfully!'
    else:
       msg = 'Incorrect username/password' return
       render_template('login.html',msg=msg)
@app.route('/register', methods=['GET', 'POST']) def
register():
  if
       request.method
                               'POST':
                       ==
                                          username
    request.form['username'] email = request.form['email']
    password = request.form['password'] sql = "SELECT *
                               username
                    WHERE
    FROM
             users
    ibm_db.prepare(conn, sql) ibm_db.bind_param(stmt,
    1, username) ibm_db.execute(stmt)
    account =
    ibm_db.fetch_assoc(stmt)
    print(account) if account:
       msg = "Account already exists!"
    elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
```

```
msg = "Invalid email address"
    elif not re.match(r'[A-Za-z0-9+', username):
       msg = "name must contain characters and numbers" else:
       insert_sql = "INSERT INTO users VALUES(?, ?,
       ?)" prep_stmt = ibm_db.prepare(conn.insert_sql)
       ibm_db.bind_param(prep_stmt, 1, username)
       ibm_db.bnd_param(prep_stmt, 2, email)
       ibm_db.bind_param(prep_stmt, 3, password)
       ibm_db.execute(prep_stmt) msg = "You have
       successfully registered"
  elif request.method == "POST":
       msg == "Please fill out the form"
  return render_template('register.html', msg=msg)
@app.route('/dashboard')
def dash():
  return render_template('dashboard.html')
@app.route('/apply', methods=['GET","POST'])
def app(): msg = ' ' if request.method ==
"POST": username = request.form['username']
email = request.form['email'] qualification =
request.form['qualification']
                                 skills
request.form['skills'] jobs = request.form['s']
```

```
ibm_db.prepare(conn,sql)
stmt
            =
                             1,
ibm_db.bind_param(stmt,
                                    username)
ibm db.execute(stmt)
                             account
ibm_db.fetch_assoc(stmt) print(account)
if account():
       msg = "there is only 1 job position" return
       render_template('apply.html',
                                         msg=msg)
       insert_sql = "INSERT INTO job VALUES(?,
       ?, ?, ?)" prep_stmt = ibm_db.prepare(conn,
       insert_sql) ibm_db.bind_param(prep_stmt, 1,
       username) ibm_db.bind_param(prep_stmt, 2,
                 ibm db.bind param(prep stmt,
       email)
                                                  3.
       qualification) ibm_db.bind_param(prep_stmt,
       4, skills) ibm_db.bind_param(prep_stmt, 5,
       jobs) ibm_db.execute(prep_stmt) msg = "You
                successfully
                               applie
                                         for
                                               iob"
       have
       session['Loggedin'] = True
     TEXT = "Hello user, a new application for job position + job
+ is requested"
   elif request.method == "POST": msg
     = "Please fill out the form"
```

return render\_template('register.html', msg=msg)

```
@app.route('/display') def
display():
   print
   session["username"], session['id']
                                              cursor
                                                          =
   mysql.connection.cursor()
   cursor.execute('SELECT*FROM job WHERE userid=%s',
(session['id'],))
                    account
   cursor.fetchone()
  print("accountdisplay", account)
return render_template('display.html',account=account)
@app.route('/logout')
def logout():
   session.pop('loggedin',None)
   session.pop('id',None)
   session.pop('username',None)
   return render_template('home.html')
if __name__== '__main___': app.run(host
   ='0.0.0.0'
```