

# ASSIGNMENT – II

**1, 2: - Create user table with email USERNAME ,ROLL\_NO, PASSWORD and perform insert ,update and delete.**

## QUERY:

```
create table user1(username varchar(255),email varchar(255), roll_no
varchar(255),password varchar(255));
insert into user1
values('Sneka','sneka001@gmail.com','85','sffgh@01'),('lavanya','lavanya@gmail.c
om','53','Zxsfm@02'),('chotu','chotu@gmail.com','67','Qwerty@07'),('loki','loki@gmai
l.com','54','loki@123');
select * from user1;
update user1 set email='sneka030@gmail.com' where roll_no='85';
insert into user1 values('nishanth','nishanth@gmail.com','025','Omfs@025');
delete user1 where roll_no='67';
select * from user1;
```

The screenshot displays the IBM Db2 on Cloud interface. The main window shows a SQL script titled 'Untitled - 1' with the following content:

```
1 create table user1(username varchar(255),email varchar(255), roll_no varchar(255),password varchar(255));
2 insert into user1 values('Sneka','sneka001@gmail.com','85','sffgh@01'),('lavanya','lavanya@gmail.com','53','Zxsfm@02');
3 select * from user1;
4 update user1 set email='sneka030@gmail.com' where roll_no='85';
5 insert into user1 values('nishanth','nishanth@gmail.com','025','Omfs@025');
6 delete user1 where roll_no='67';
7 select * from user1;
```

The 'History' tab shows the execution results for the script:

Script	Date	Status	Runtime
Untitled - 1	Oct 13, 2022 3:49:11 PM	6 1	0.062 s
create table user1(username varchar(255),email varchar(255), roll_no varchar(255),password varchar(255));			0.017 s
insert into user1 values('Sneka','sneka001@gmail.com','85','sffgh@01'),('lavanya','lavanya@gmail.com','53','Zxsfm@02');			0.005 s
select * from user1;			0.015 s
update user1 set email='sneka030@gmail.com' where roll_no='85';			0.005 s

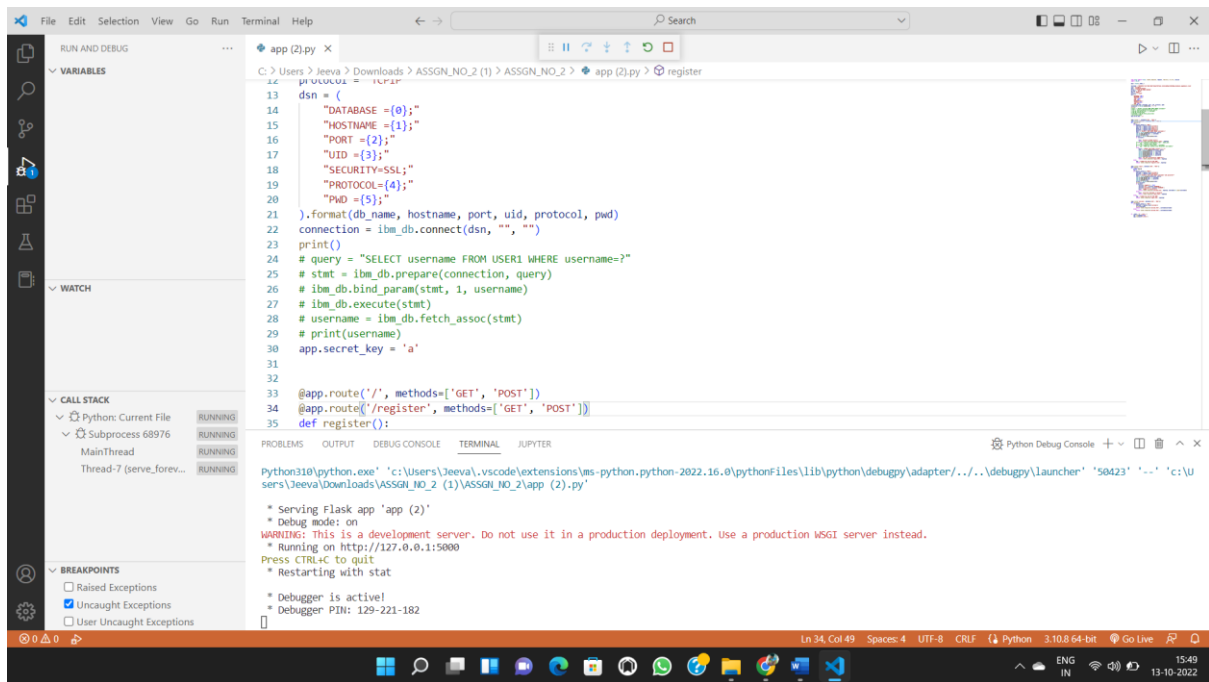
### 3. Connect python to db2

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import db
import db2
import re
```

```
hostname = '6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud'
uid = 'SCC60467'
pwd = '81Vc7QTT0qX7oyIc'
driver = "{IBM DB2 ODBC DRIVER}"
db_name = 'Bludb'
port = '30376'
protocol = 'TCPIP'
dsn = (
    "DATABASE={0};"
    "HOSTNAME={1};"
    "PORT={2};"
    "UID={3};"
    "SECURITY=SSL;"
    "PROTOCOL={4};"
    "PWD={5};"
).format(db_name, hostname, port, uid, protocol, pwd)
connection = ibm_db.connect(dsn, "", "")
print(dsn)
try:
```

```
    print("Connecting to db2.....")
    db2 = ibm_db.connect(dsn, "", "")
    print()
    print("Connected to database")
    print("Connection Successful!!!")
```

```
except Exception as exception:
    print("unable to connect ", exception)
```



## 4) ACCESS LOGIN WITH CONNTING TO DATABASE

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

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

```
hostname = '6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.clogj3sd0tgtu0lqde00.databases.appdomain.cloud'
uid = 'SCC60467'
pwd = '81Vc7QTToqX7oyIc'
driver = "{IBM DB2 ODBC DRIVER}"
db_name = 'Bludb'
port = '30376'
protocol = 'TCPIP'
dsn = (
    "DATABASE={0};"
    "HOSTNAME={1};"
    "PORT={2};"
    "UID={3};"
    "SECURITY=SSL;"
    "PROTOCOL={4};"
    "PWD={5};"
).format(db_name, hostname, port, uid, protocol, pwd)
connection = ibm_db.connect(dsn, "", "")
print()
```

```

# query = "SELECT username FROM USER1 WHERE username=?"
# stmt = ibm_db.prepare(connection, query)
# ibm_db.bind_param(stmt, 1, username)
# ibm_db.execute(stmt)
# username = ibm_db.fetch_assoc(stmt)
# print(username)
app.secret_key = 'a'

@app.route('/', methods=['GET', 'POST'])
@app.route('/register', methods=['GET', 'POST'])
def register():
    msg = " "
    if request.method == 'POST':
        username = request.form['username']
        email_id = request.form['email_id']
        phone_no = request.form['phone_no']
        password = request.form['password']
        query = "SELECT * FROM USER1 WHERE username=?;"
        stmt = ibm_db.prepare(connection, query)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if (account):

            msg = "Account already exists!"
            return render_template('register.html', msg=msg)
        # elif not re.match(r'^@]+@[^@]+\.[^@]+', email_id):
        #     msg = "Invalid email address"
        # elif not re.match(r'[A-Za-z0-9+]', username):
        #     msg = "Name must contain only characters and numbers"
        else:
            query = "INSERT INTO USER1 values(?,?,?,?)"
            stmt = ibm_db.prepare(connection, query)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email_id)
            ibm_db.bind_param(stmt, 3, phone_no)
            ibm_db.bind_param(stmt, 4, password)
            ibm_db.execute(stmt)
            msg = 'You have successfully Logged In!!'
            return render_template('login.html', msg=msg)
    else:
        msg = 'PLEASE FILL OUT OF 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']
    query = "select * from user1 where username=? and password=?"
    stmt = ibm_db.prepare(connection, query)
    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']
        session['username'] = account['USERNAME']
        msg = 'Logged in Successfully'
        return render_template('welcome.html', msg=msg,
username=str.upper(username))
    else:
        msg = 'Incorrect Username or Password'
        return render_template('login.html', msg=msg)
else:
    msg = 'PLEASE FILL OUT OF THE FORM'
    return render_template('login.html', msg=msg)

@app.route('/welcome', methods=['GET', 'POST'])
def welcome():
    if request.method == 'POST':
        username = request.form['username']
        print(username)
        return render_template('welcome.html', username=username)
    else:
        return render_template('welcome.html', username=username)

if __name__ == "__main__":
    app.run(debug=True)
    app.run(host='0.0.0.0')

```