

Assignment - 2 Personal Expense Tracker

Paulsan S
PNT2022PMID19316

Creation of Table:

The screenshot shows the IBM Db2 on Cloud interface. The 'Tables' tab is selected, displaying a list of tables in the 'NLG66799' schema: 'USER' and 'USERS'. The 'Table definition' panel for the 'USER' table is open, showing the following columns:

Name	Data type	Nullable	Length	Scale
EMAIL	VARCHAR	Y	32	0
USERNAME	VARCHAR	Y	32	0
roll number	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

A 'View data' button is visible at the bottom of the table definition panel.

Insertion into USER Table:

The screenshot shows the IBM Db2 on Cloud interface with the SQL editor open. The editor contains the following SQL statement:

```
1 insert into USER values('user1@gmail.com', 'user1', '0001', 'pwd1');
2 insert into USER values('user2@gmail.com', 'user2', '0002', 'pwd2');
3 insert into USER values('user3@gmail.com', 'user3', '0003', 'pwd3');
4 insert into USER values('user4@gmail.com', 'user4', '0004', 'pwd4');
```

The 'Run all' button is visible in the top right corner of the editor. The left sidebar shows the 'Data objects' tree with the 'NLG66799' schema expanded, showing 'Tables' and 'USERS'.

IBM Db2 on Cloud

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

NLG66799.USER

Back

Export to CSV

EMAIL	USERNAME	roll number	PASSWORD
user1@gmail.com	user5	0001	pwd1
user2@gmail.com	user2	0002	pwd2
user3@gmail.com	user3	0003	pwd3
user4@gmail.com	user4	0004	pwd4

Deleting from USER Table:

IBM Db2 on Cloud

Data objects My script

Filter objects

NLG66799

- Tables
 - USER
 - USERS
- Views
- MQTs
- Aliases
- Nicknames

*Untitled - 1

Beta Classic

Run all

```
1 delete from user where username='user5';
```

History

Find history

EMAIL	USERNAME	roll number	PASSWORD
user2@gmail.com	user2	0002	pwd2
user3@gmail.com	user3	0003	pwd3
user4@gmail.com	user4	0004	pwd4

Connecting to IBM DB2 From Python:

```
conn = ibm_db.connect(
    "DATABASE=bludb;\
    HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;\
    PORT=30699;\
    Security=SSL;\
    SSLServerCertificate=DigiCertGlobalRootCA.crt;\
    UID=nlg66799;\
    PWD=CXtQLAGZ06fD0fhC;", "", "")
```

Source Code

App.py

```
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=54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.da
tabases.appdomain.cloud;\
PORT=30699;\
Security=SSL;\
SSLServerCertificate=DigiCertGlobalRootCA.crt;\
UID=nlq66799;\
PWD=CXtQLAGZ06fD0fhC;","","")
```

```
@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 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>
<h2 style="color:blue">{{msg}}</h2>
<form action="/" method="POST">
<fieldset>
    <legend>User Information</legend>
<h2>Username:<h2> <input type="text" name="username"
placeholder="Enter username
"><br/><br/>
<h2>Password:<h2> <input type="password" name="password"><br/><br/>
<h2>Email:<h2> <input type="email" name="email" placeholder="Enter
email"><br/><br/>
<h2>Roll number:<h2> <input type="text" name="rollno"
placeholder="Enter roll
number"><br/><br/>
<input type="submit" value="Register">
</fieldset>
</form>
</body>
</html>

```

login.html

```

<html>
<body>
<h2 style="color:blue">{{msg}}</h2>

```

```
<form action="/login" method="POST">
  <fieldset>
    <legend>User Login:</legend>
    <h2>Username:</h2> <input type="text" name="username"
placeholder="Enter
username"><br/><br/>
    <h2>Password:</h2> <input type="password" name="password"
placeholder="*****"><br/><br/>
    <input type="submit" value="Login">
  </fieldset>
</form>
</body>
</html>
```

welcome.html

```
<h1 style="color:blue">Welcome {{username}}!</h1>
<h2>{{msg}}</h2>
```

Output:

register.html

Please fill the form

User Information

Username:

Password:

Email:

Roll number:

login.html (correct login)

Login

User Login:

Username:

Password:

Login

welcome.html

Welcome usernew !

Logged in successfully

IBM Db2 on Cloud

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

NLG66799.USER

Back

Export to CSV

EMAIL	USERNAME	roll number	PASSWORD
user2@gmail.com	user2	0002	pwd2
user3@gmail.com	user3	0003	pwd3
user4@gmail.com	user4	0004	pwd4
usernew@gmail.com	usernew11	usernew	usernew

login.html (incorrect login)

Login

User Login:

Username:

Password:

Login

Incorrect username/password

User Login:

Username:

Enter username

Password:

Login