

ASSIGNMENT – 2

Student Name	Santhosh S
Student Roll number	311519104051
Team ID	PNT2022TMID27815

1. CREATE USER TABLE WITH USER WITH EMAIL, USERNAME, ROLL NUMBER, PASSWORD

TABLE CREATION:

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, and a table named 'USER' is being created in the 'YXK79848' schema. The table definition is as follows:

Name	Data type	Nullable	Length	Scale
EMAIL	VARCHAR	N	32	0
USERNAME	VARCHAR	N	32	0
ROLL_NO	INTEGER	N		0
PASSWORD	VARCHAR	N	32	0

The interface also shows a 'New table' button and a 'View data' button. The table definition panel indicates 'Approximate 2 rows (32.0 KB)' and 'Updated on 2022-10-30 13:18:16'.

The screenshot shows the IBM Db2 on Cloud web interface with the SQL editor open. The SQL script being executed is:

```
1 insert into USER values('user@gmail.com','user',10,'user');
2 insert into USER values('Santhosh@gmail.com','Santhosh',51,'santhosh@123');
3 insert into USER values('Sandy@gmail.com','user',52,'Sandy@123');
```

The 'Run all' button is highlighted. Below the editor, the 'History' tab is selected, showing the execution history of the script:

Script	Date	Status	Runtime
Untitled - 2	Oct 30, 2022 4:06:19 PM	3	0.020 s
insert into USER values('user@gmail.com','user',10,'user')		✓	0.008 s
insert into USER values('Santhosh@gmail.com','Santhosh',51,'san...		✓	0.006 s
insert into USER values('Sandy@gmail.com','user',52,'Sandy@123'...		✓	0.006 s

Service Details - IBM Cloud | IBM Db2 on Cloud | (1) WhatsApp

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Ffb61a597079f4fa39a69c029f1fad4...

IBM Db2 on Cloud

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

YXK79848.USER

Back

Export to CSV

EMAIL	USERNAME	ROLL_NO	PASSWORD
Sandy@gmail.com	user	52	Sandy@123
Santhosh@gmail.com	Santhosh	51	santhosh@123
user@gmail.com	user	10	user

30°C Haze 16:07 30-10-2022

2. PERFORM UPDATE, DELETE QUERIES WITH USER TABLE

UPDATE QUERY:

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

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Ffb61a597079f4fa39a69c029f1fad4...

IBM Db2 on Cloud

Data objects | Saved objects

Filter objects

YXK79848

Tables

USER

Views

MQTs

Aliases

Nicknames

*Untitled - 2

```

1 insert into USER values('user@gmail.com','user',10,'user');
2 insert into USER values('Santhosh@gmail.com','Santhosh',51,'santhosh@123');
3 insert into USER values('Sandy@gmail.com','user',52,'Sandy@123');
4
5 update USER set Username='Sandy' where Email='Sandy@gmail.com';

```

Syntax assistant

Run selected

History

Find history

Script	Date	Status	Runtime
Untitled - 2	Oct 30, 2022 4:14:43 PM	1	0.008 s
update USER set Username='Sandy' where Email='Sandy@gmail.com'			0.008 s
Untitled - 2	Oct 30, 2022 4:06:19 PM	3	0.020 s
insert into USER values('user@gmail.com','user',10,'user')			0.008 s
insert into USER values('Santhosh@gmail.com','Santhosh',51,'s...			0.006 s

30°C Haze 16:15 30-10-2022

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

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Ffb61a597079f4fa39a69c029f1fad4...

IBM Db2 on Cloud

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

YXK79848.USER

Back

Export to CSV

EMAIL	USERNAME	ROLL_NO	PASSWORD
Sandy@gmail.com	Sandy	52	Sandy@123
Santhosh@gmail.com	Santhosh	51	santhosh@123
user@gmail.com	user	10	user

30°C Haze 16:15 30-10-2022

DELETE QUERY:

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

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Ffb61a597079f4fa39a69c029f1fad4...

IBM Db2 on Cloud

Data objects | Saved objects

Filter objects

YXK79848

Tables

USER

Views

MQTs

Aliases

Nicknames

*Untitled - 2

Syntax assistant

Run selected

```

1 insert into USER values('user@gmail.com','user',10,'user');
2 insert into USER values('Santhosh@gmail.com','Santhosh',51,'santhosh@123');
3 insert into USER values('Sandy@gmail.com','user',52,'Sandy@123');
4
5 update USER set Username='Sandy' where Email='Sandy@gmail.com';
6
7 delete from USER where Roll_No=10;

```

History

Find history

Script	Date	Status	Runtime
Untitled - 2	Oct 30, 2022 4:17:25 PM	1	0.007 s
delete from USER where Roll_No=10			0.007 s
Untitled - 2	Oct 30, 2022 4:14:43 PM	1	0.008 s
update USER set Username='Sandy' where Email='Sandy@gmail.com'			0.008 s
Untitled - 2	Oct 30, 2022 4:06:19 PM	3	0.020 s
insert into USER values('user@gmail.com','user',10,'user')			0.008 s

30°C Haze 16:17 30-10-2022

IBM Db2 on Cloud

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

YXK79848.USER

Back

Export to CSV

EMAIL	USERNAME	ROLL_NO	PASSWORD
Sandy@gmail.com	Sandy	52	Sandy@123
Santhosh@gmail.com	Santhosh	51	santhosh@123

3. CONNECT PYTHON TO DB2

CONNECTING TO DB:

ass.py

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

```
app = Flask(__name__)
app.secret_key = 'a'
```

```
conn = ibm_db.connect("DATABASE=bludb; HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt; UID=yxk79848; PWD=dgzflUCbtRBACahK;", "", "")
```

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

```
    return render_template('reg2.html')
```

```
@app.route('/login', methods=["GET", "POST"])
```

```
def login():
```

```
    global userid
    msg = ""
```

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

```
        username = request.form['uname']
```

```
        password = request.form['pwd']
```

```
        sql = "SELECT * FROM Users 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 redirect(url_for('welcome', username=username))
else:
    msg = "Incorrect Username/Password"
    return render_template('login.html', msg=msg)

```

```
@app.route('/reg2', methods=["GET", "POST"])
```

```
def registration():
```

```
    msg = " "
```

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

```
        username = request.form['uname']
```

```
        email = request.form['email']
```

```
        password = request.form['pwd']
```

```
        rollno = request.form['rollno']
```

```
        sql = "SELECT * FROM USERS WHERE USERNAME=?"
```

```
        stmt = 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'^[^\s@]+@[^\s@]+\.[^\s@]+', email):
```

```
            msg = "Invalid Email Address."
```

```
        elif not re.match(r'[A-Za-z0-9]+', username):
```

```
            msg = "Username must contain only alphabets and numbers."
```

```
        else:
```

```
            insert_sql = "INSERT INTO USERS VALUES(?,?,?,?)"
```

```
            prep_stmt = ibm_db.prepare(conn, insert_sql)
```

```
            ibm_db.bind_param(prepare_stmt, 1, email)
```

```
            ibm_db.bind_param(prepare_stmt, 2, username)
```

```
            ibm_db.bind_param(prepare_stmt, 3, rollno)
```

```
            ibm_db.bind_param(prepare_stmt, 4, password)
```

```
            ibm_db.execute(prepare_stmt)
```

```
            msg = "You have successfully registered."
```

```
        return render_template('login.html', msg=msg)
```

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

```
        msg = "Please fill out the form."
```

```
        return render_template('reg2.html', msg=msg)
```

```
@app.route('/welcome/<username>')
```

```
def welcome(username):  
    return "Welcome %s!" % username
```

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

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

reg.html (Registration Page):-

```
<html>  
<body>  
    <center>  
        <h3><b>REGISTRATION FOR PLASMA DONATION</b></h3>  
        <form action = "http://localhost:5000/reg2" method = "POST">  
            <p>Enter Register No.: <input type = "text" name="rollno"/></p>  
            <p>Enter Email ID: <input type = "text" name="email"/></p>  
            <p>Enter Username: <input type = "text" name="uname"/></p>  
            <p>Enter Password: <input type = "password" name="pwd"/></p>  
            <p><input type = "submit" value="SUBMIT"/></p>  
        </form>  
    </center>  
</body>  
</html>
```

login.html (Login Page):-

```
<html>  
<body>  
    <center>  
        <h3><b>LOGIN</b></h3>  
        <form action = "http://localhost:5000/login" method = "POST">  
            <p>Enter Username: <input type = "text" name="uname"/></p>  
            <p>Enter Password: <input type = "password" name="pwd"/></p>  
            <p><input type = "submit" value="SUBMIT"/></p>  
        </form>  
    </center>  
</body>  
</html>
```

Registration page:

← → ↻ ⓘ 127.0.0.1:5000

REGISTER FOR PLASMA DONATION

Enter Register No.:

Enter Email ID:

Enter Username:

Enter Password:

Login Page:

← → ↻ ⓘ localhost:5000/reg2

LOGIN

Enter Username:

Enter Password:

Welcome Page:

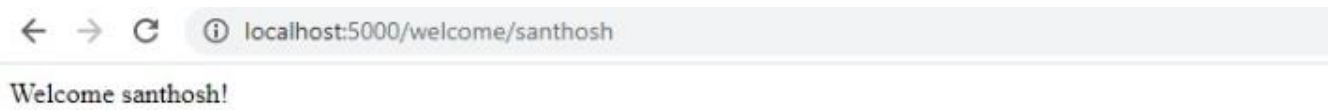


Table Insertion:

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

RRJ92864.USERS

Back

Export to CSV

EMAIL	USERNAME	ROLLNO	PASSWORD
santhosh@gmail.com	santhosh	4051	santhosh

Activate Windows
Go to PC settings to activate Windows.

