

Assignment -2

Student Name	ABDUL ADHIL K
Student Roll Number	7376191CS102
Team ID	PNT2022TMID01433

Question-1:

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

Solution

Creating a users table in Ibm cloud

The screenshot displays the IBM Db2 on Cloud console interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is active, showing a search bar and a 'Refresh' button. Below the search bar, a table lists the existing tables:

Name	Schema	Properties
USERS	WXS77796	...

The 'USERS' table is selected. To the right, the 'Table definition' panel shows the schema details for the 'USERS' table, which has approximately 1 row (32.0 KB) and was updated on 2022-10-30 13:18:03. The table structure is as follows:

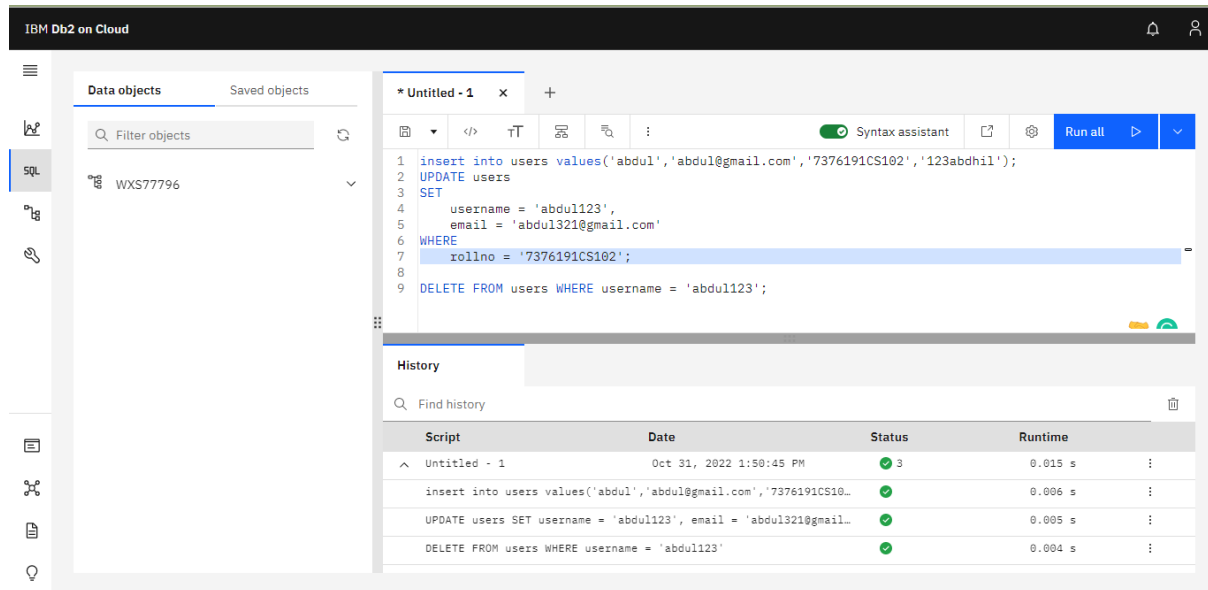
Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	Y	32	0
EMAIL	VARCHAR	Y	32	0
ROLLNO	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

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

Question-2:

Perform UPDATE,DELETE Queries with user table

Solution



The screenshot shows the IBM Db2 on Cloud interface. On the left, there's a sidebar with 'Data objects' and 'Saved objects' tabs. The 'Data objects' tab is active, showing a search bar and a list of objects, including 'WXS77796'. The main area displays a SQL editor with the following queries:

```
1 insert into users values('abdul','abdul@gmail.com','7376191CS102','123abdhi1');
2 UPDATE users
3 SET
4     username = 'abdul123',
5     email = 'abdul321@gmail.com'
6 WHERE
7     rollno = '7376191CS102';
8
9 DELETE FROM users WHERE username = 'abdul123';
```

Below the editor, there's a 'History' section with a table showing the execution history of the scripts.

Script	Date	Status	Runtime
Untitled - 1	Oct 31, 2022 1:50:45 PM	3	0.015 s
insert into users values('abdul','abdul@gmail.com','7376191CS102')		✓	0.006 s
UPDATE users SET username = 'abdul123', email = 'abdul321@gmail.com'		✓	0.005 s
DELETE FROM users WHERE username = 'abdul123'		✓	0.004 s

Question-3:

Connect python code to db2.

```
app.py x
app.py > ...
1 from flask import Flask, render_template, request, redirect, url_for, session
2 import ibm_db
3 import re
4
5 app = Flask(__name__)
6
7 app.secret_key = 'a'
8
9 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=fb88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd8gtu@lqde00.databases.appdomain.cloud;P
10
11
12 @app.route('/', methods = ['GET', 'POST'])
13 def register():
14     msg = ''
15     if request.method == 'POST':
16         username = request.form['username']
17         email = request.form['email']
18         rollno = request.form['rollno']
19         password = request.form['password']
20         sql = "SELECT * FROM users WHERE username =?"
21         stmt = ibm_db.prepare(conn, sql)
22         ibm_db.bind_param(stmt, 1, username)
23         ibm_db.execute(stmt)
24         account = ibm_db.fetch_assoc(stmt)
25         print(account)
26         if account:
27             msg = 'Account already exists !'
28         elif not re.match(r'[^@]+\@[^@]+\.[^@]+', email):
29             msg = 'Invalid email address !'
30         elif not re.match(r'[A-Za-z0-9]+', username):
31             msg = 'name must contain only characters and numbers !'
32         else:
33             insert_sql = "INSERT INTO users VALUES (?, ?, ?, ?)"
34             prep_stmt = ibm_db.prepare(conn, insert_sql)
35             ibm_db.bind_param(prepare_stmt, 1, username)
36             ibm_db.bind_param(prepare_stmt, 2, email)
37             ibm_db.bind_param(prepare_stmt, 3, rollno)
38             ibm_db.bind_param(prepare_stmt, 4, password)
39             ibm_db.execute(prepare_stmt)
40             msg = 'You have successfully registered !'
41         elif request.method == 'POST':
42             msg = 'Please fill out the form !'
43         return render_template('register.html', msg = msg)
44
```

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

Registration Page:

Registration page

Username:

Email:

Roll number:

Password:

already have an account ? please login [login!](#)

Code:

```
templates > register.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
5    <link rel="stylesheet" href="../../static/CSS/style.css" />
6    <title>Registration page</title>
7  </head>
8  <body>
9    <form action="http://localhost:5000/" method="post">
10     <div class="container">
11       <h1>Registration page</h1>
12       <hr />
13
14       <div class="msg">{{ msg }}</div>
15
16       <label for="username"><b>Username:</b></label>
17       <input
18         type="text"
19         placeholder="Enter your Username"
20         name="username"
21         id="username"
22         required
23       />
24
25       <label for="email"><b>Email:</b></label>
26       <input
27         type="text"
28         placeholder="Enter your Email"
29         name="email"
30         id="email"
31         required
32       />
33
34       <label for="rollno"><b>Roll number:</b></label>
35       <input
36         type="text"
37         id="rollno"
38         name="rollno"
39         placeholder="Enter your Rollno"
40       />
41
42       <label for="password"><b>Password:</b></label>
43       <input
44         type="password"
45         id="password"
46         name="password"
47         placeholder="Enter your password"
48       />
49
50       <button type="submit" class="registerbtn">Register</button>
51     </div>
52   </form>
53 </body>
54 </html>
```

Login Page:

Login Form

Username:

Password:

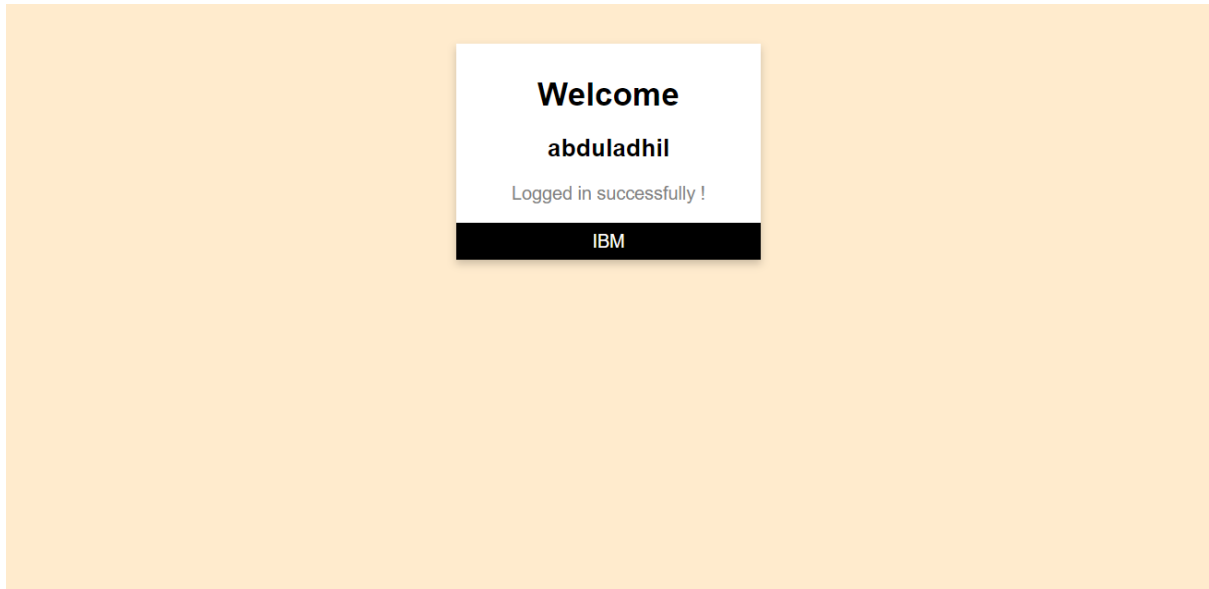
Login

Don't have an account yet? Click here to [register!](#)

Code:

```
templates > login.html > html > body > div.container > div.login.text-center.mt-5 > form
3   <head>
4   <meta charset="UTF-8" />
5   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <title>Login Page</title>
8
9   <link rel="stylesheet" href="../../static/CSS/style.css" />
10
11  </head>
12  <body>
13    <div class="container">
14      <div class="login text-center mt-5">
15        <h1>Login Form</h1>
16        <hr>
17        <form action="/login" method="post">
18          <div class="msg">{{ msg }}</div>
19          <label for="username"><b>Username:</b></label>
20          <input
21            type="text"
22            name="username"
23            placeholder="Enter Your Username"
24            id="username"
25            required
26          />
27
28          <label for="password"><b>Password:</b></label>
29          <input
30            type="password"
31            name="password"
32            placeholder="Enter Your Password"
33            id="password"
34            required
35          />
36
37          <button type="submit" id="button" class="registerbtn">Login</button>
38        </form>
39      </div>
40
41      <div class="note mt-3 text-center">
42        <p>
43          Don't have an account yet? Click here to <a href="/">register! </a>
44        </p>
```

Welcome Page:



Code:

```
templates > display.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <link rel="stylesheet" href="../static/CSS/display.css" />
8      <title>Welcome Page</title>
9  </head>
10 <body style="background-color: #blanchedalmond;">
11
12 <div class="card">
13     <h1>Welcome</h1>
14     <h2>{{name}}</h2>
15     <p class="title">{{ msg }}</p>
16     <p></p>
17
18     <p><button>IBM</button></p>
19 </div>
20
21 </body>
22 </html>
23
```