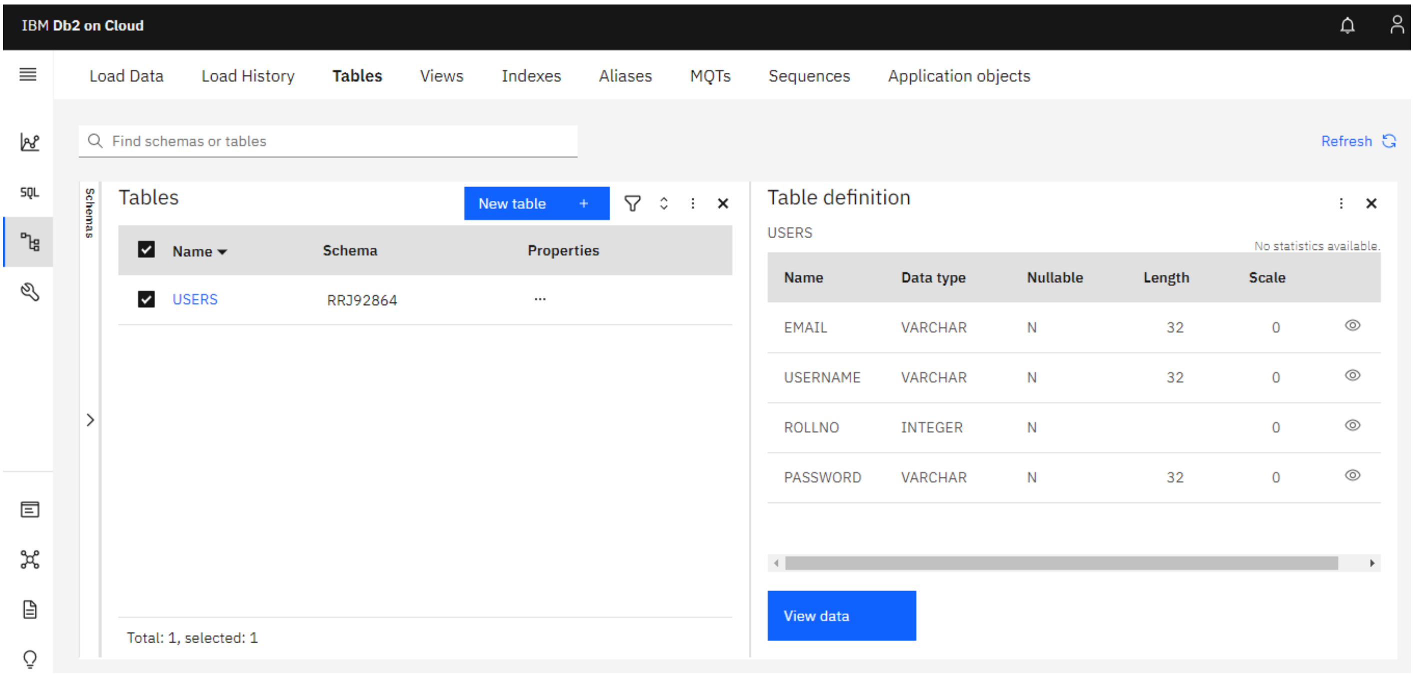
ASSIGNMENT – 2

Student Name	Jaimugil C
Student Roll number	311519104023
Team ID	PNT2022TMID27815

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

TABLE CREATION:



Tables

☑ Name	Schema	Properties
☑ USERS	RRJ92864	...

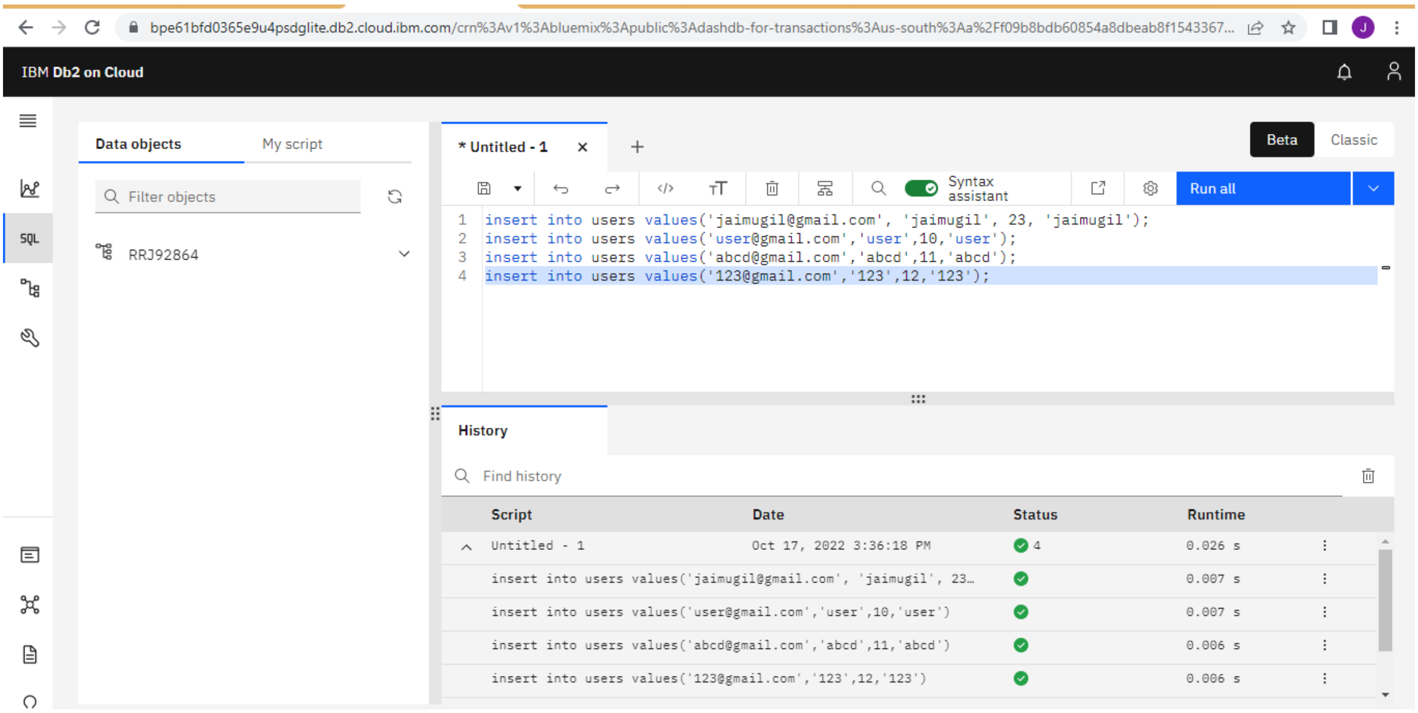
Total: 1, selected: 1

Table definition

USERS

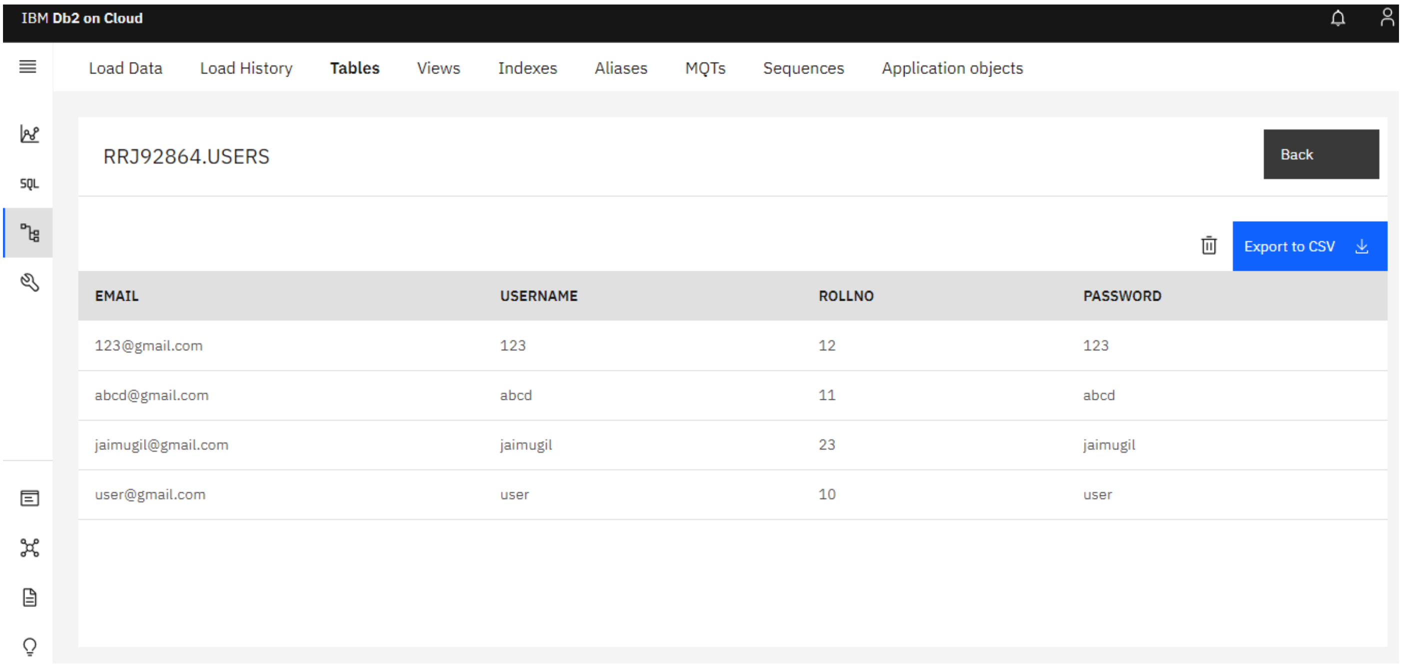
No statistics available.

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


bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Ff09b8bdb60854a8dbeab8f1543367...
IBM Db2 on Cloud
Data objects
My script
Filter objects
RRJ92864
* Untitled - 1
Beta
Classic
Syntax assistant
Run all
SQL
History
Find history

```
insert into users values('jaimugil@gmail.com', 'jaimugil', 23, 'jaimugil');
insert into users values('user@gmail.com','user',10,'user');
insert into users values('abcd@gmail.com','abcd',11,'abcd');
insert into users values('123@gmail.com','123',12,'123');
```

Script	Date	Status	Runtime
Untitled - 1	Oct 17, 2022 3:36:18 PM	✓ 4	0.026 s
insert into users values('jaimugil@gmail.com', 'jaimugil', 23…		✓	0.007 s
insert into users values('user@gmail.com','user',10,'user')		✓	0.007 s
insert into users values('abcd@gmail.com','abcd',11,'abcd')		✓	0.006 s
insert into users values('123@gmail.com','123',12,'123')		✓	0.006 s

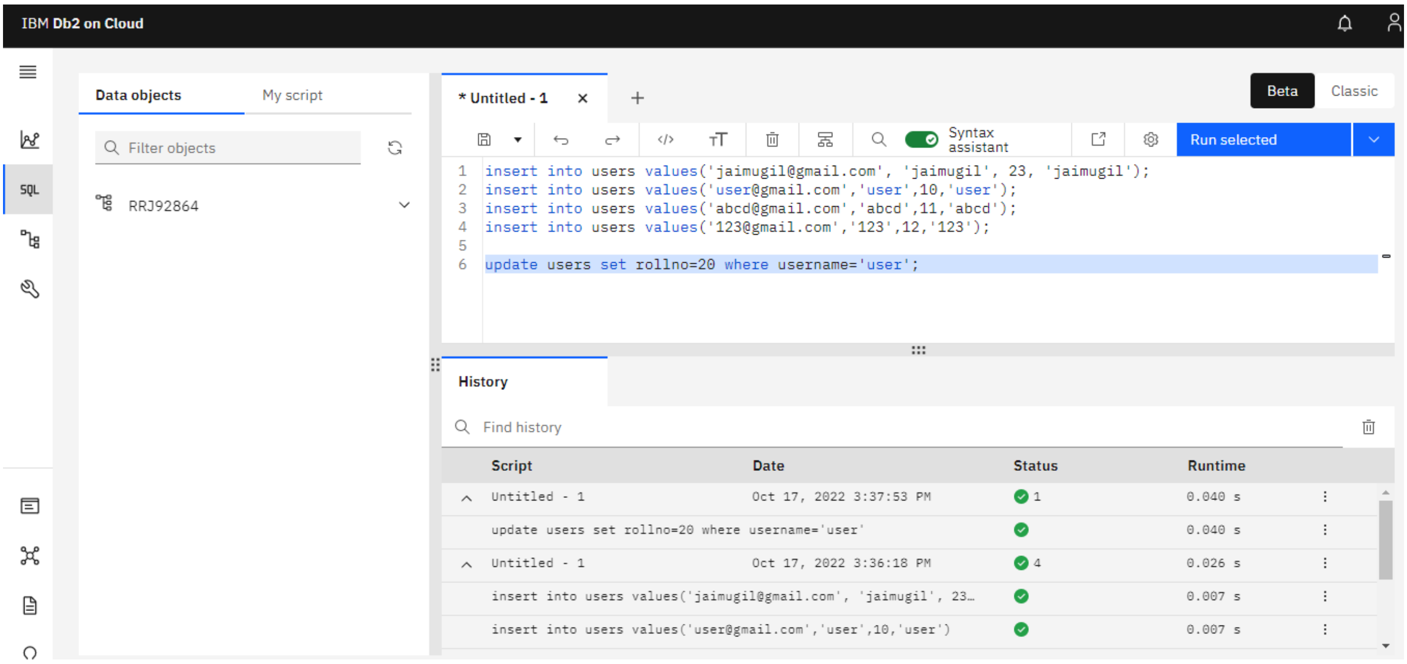


RRJ92864.USERS

EMAIL	USERNAME	ROLLNO	PASSWORD
123@gmail.com	123	12	123
abcd@gmail.com	abcd	11	abcd
jaimugil@gmail.com	jaimugil	23	jaimugil
user@gmail.com	user	10	user

2. PERFORM UPDATE, DELETE QUERIES WITH USER TABLE

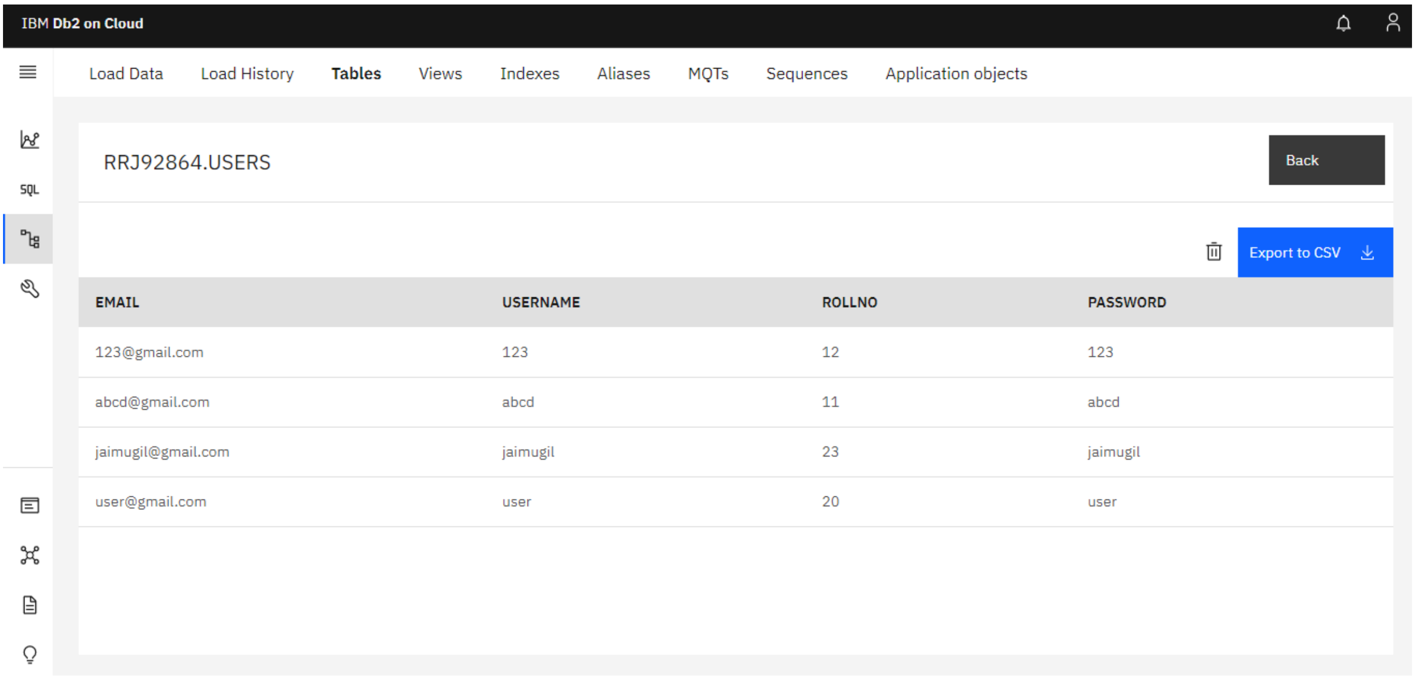
<u>UPDATE QUERY</u>:



```
insert into users values('jaimugil@gmail.com', 'jaimugil', 23, 'jaimugil');
insert into users values('user@gmail.com','user',10,'user');
insert into users values('abcd@gmail.com','abcd',11,'abcd');
insert into users values('123@gmail.com','123',12,'123');

update users set rollno=20 where username='user';
```

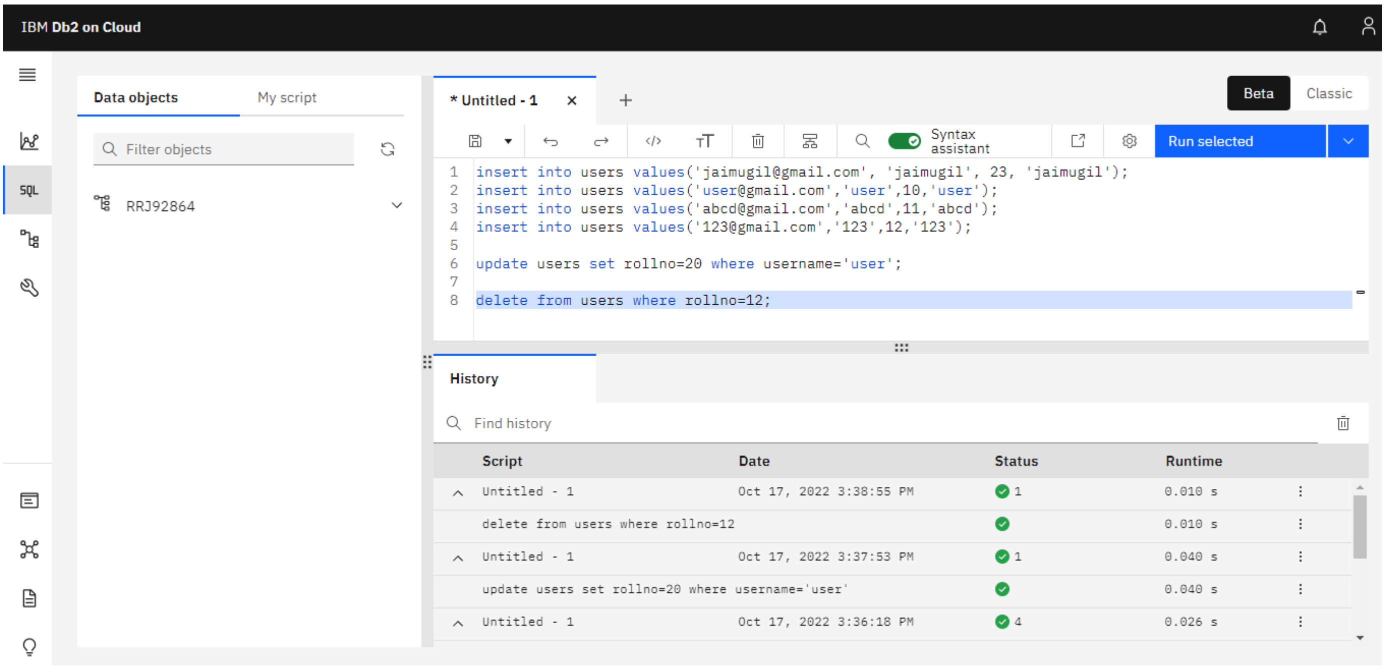
Script	Date	Status	Runtime
Untitled - 1	Oct 17, 2022 3:37:53 PM	✓ 1	0.040 s
update users set rollno=20 where username='user'		✓	0.040 s
Untitled - 1	Oct 17, 2022 3:36:18 PM	✓ 4	0.026 s
insert into users values('jaimugil@gmail.com', 'jaimugil', 23…		✓	0.007 s
insert into users values('user@gmail.com','user',10,'user')		✓	0.007 s



RRJ92864.USERS

EMAIL	USERNAME	ROLLNO	PASSWORD
123@gmail.com	123	12	123
abcd@gmail.com	abcd	11	abcd
jaimugil@gmail.com	jaimugil	23	jaimugil
user@gmail.com	user	20	user

DELETE QUERY:

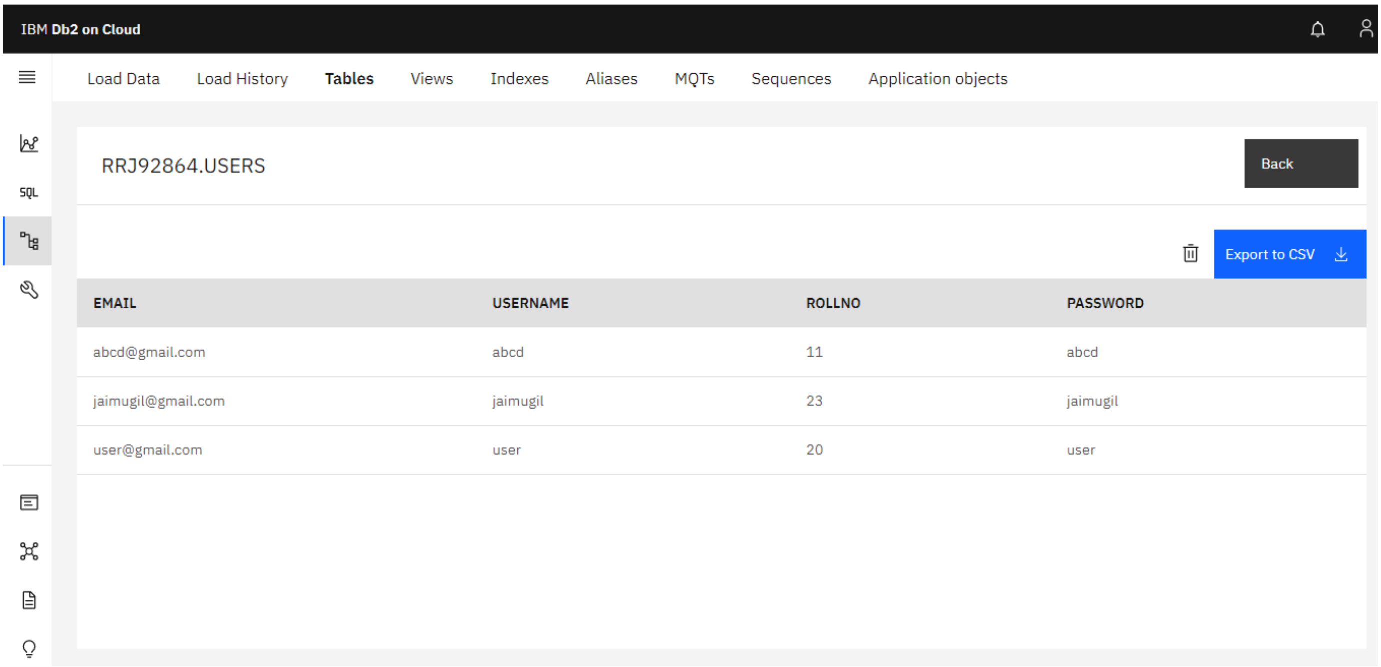


```
insert into users values('jaimugil@gmail.com', 'jaimugil', 23, 'jaimugil');
insert into users values('user@gmail.com','user',10,'user');
insert into users values('abcd@gmail.com','abcd',11,'abcd');
insert into users values('123@gmail.com','123',12,'123');

update users set rollno=20 where username='user';

delete from users where rollno=12;
```

Script	Date	Status	Runtime
Untitled - 1	Oct 17, 2022 3:38:55 PM	✓ 1	0.010 s
delete from users where rollno=12		✓	0.010 s
Untitled - 1	Oct 17, 2022 3:37:53 PM	✓ 1	0.040 s
update users set rollno=20 where username='user'		✓	0.040 s
Untitled - 1	Oct 17, 2022 3:36:18 PM	✓ 4	0.026 s



RRJ92864.USERS

EMAIL	USERNAME	ROLLNO	PASSWORD
abcd@gmail.com	abcd	11	abcd
jaimugil@gmail.com	jaimugil	23	jaimugil
user@gmail.com	user	20	user

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=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt; UID=rrj92864; PWD=maDr7SBpdojnrqgv;", '', '')
@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'[^@]+@[^@]+\.[^@]+', 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(prep_stmt, 1, email)
            ibm_db.bind_param(prep_stmt, 2, username)
            ibm_db.bind_param(prep_stmt, 3, rollno)
            ibm_db.bind_param(prep_stmt, 4, password)
            ibm_db.execute(prep_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

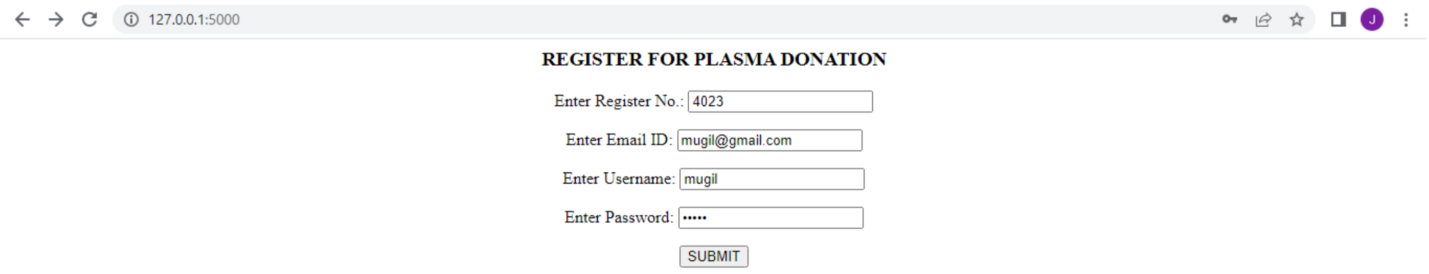
reg.html (Registration Page):-

```
    <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:



REGISTER FOR PLASMA DONATION

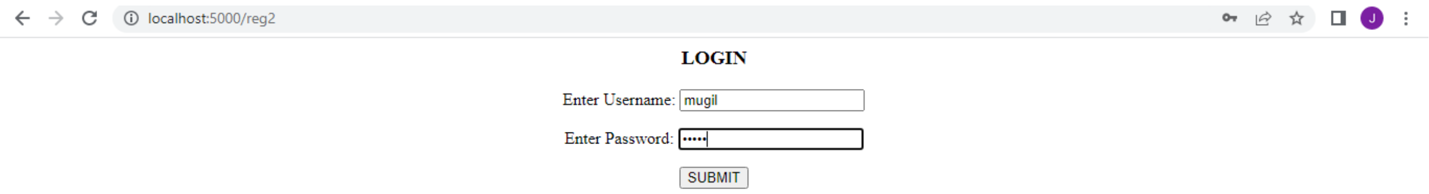
Enter Register No.: 4023

Enter Email ID: mugil@gmail.com

Enter Username: mugil

Enter Password: •••••

Login Page:

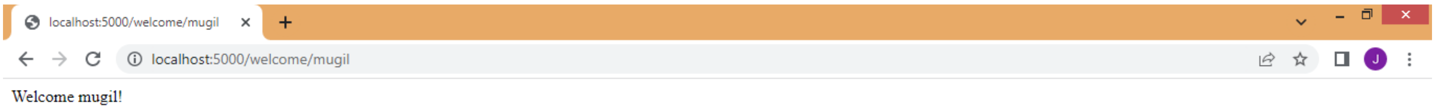


LOGIN

Enter Username: mugil

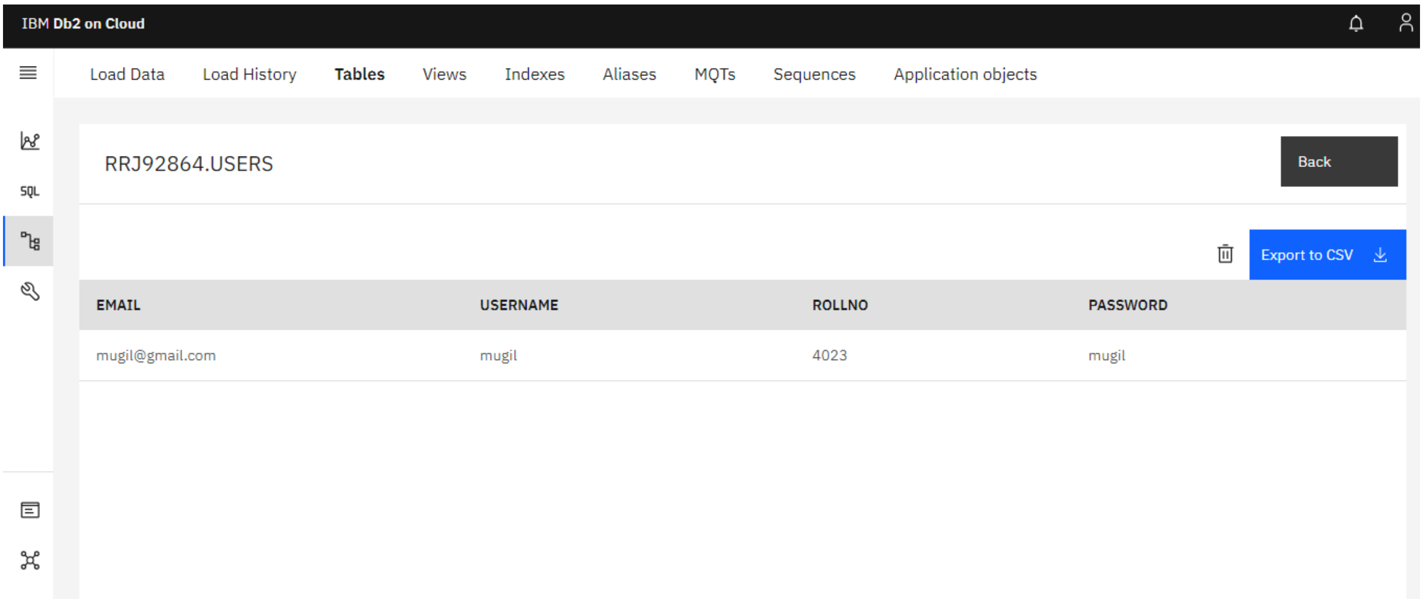
Enter Password: •••••

Welcome Page:



Welcome mugil!

Table Insertion:



RRJ92864.USERS

EMAIL	USERNAME	ROLLNO	PASSWORD
mugil@gmail.com	mugil	4023	mugil