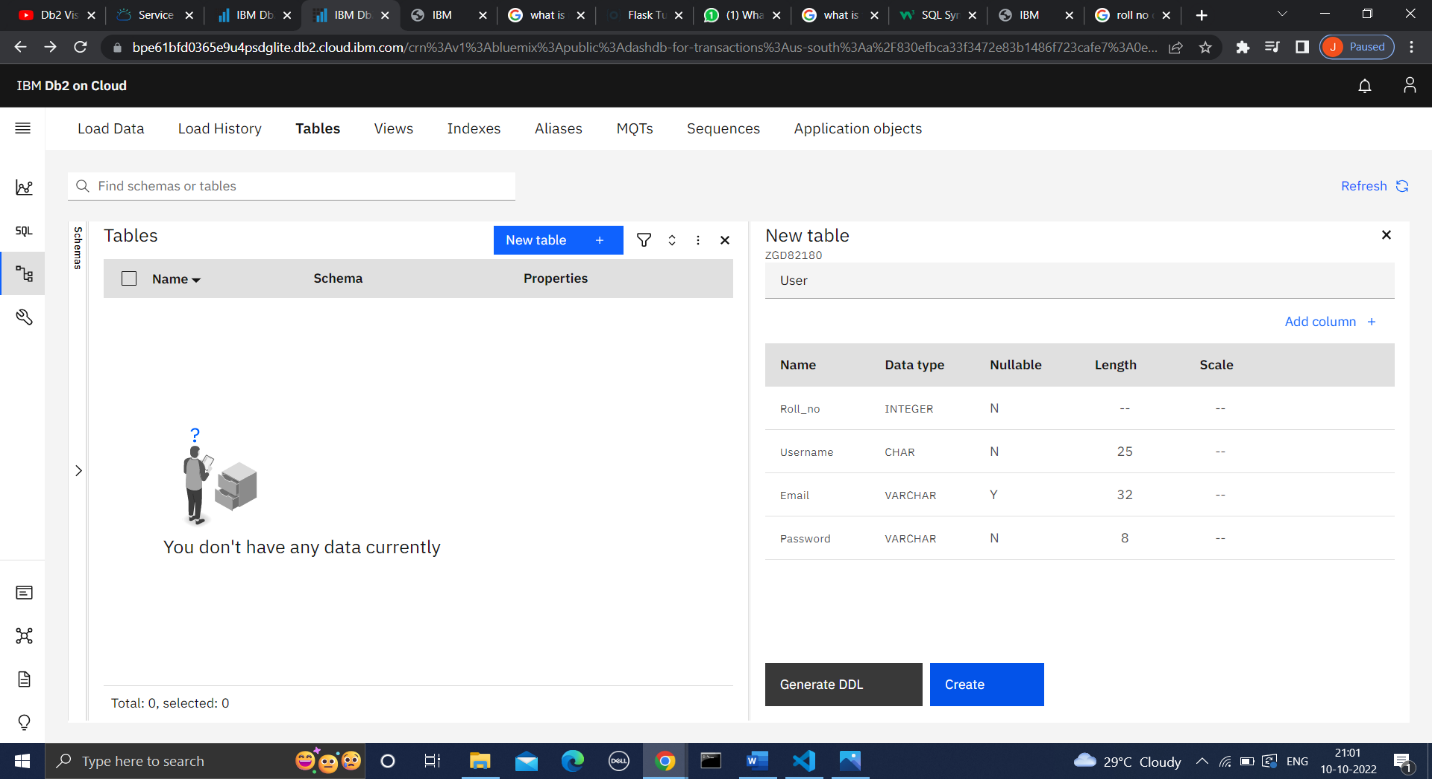
**Assignment -2**

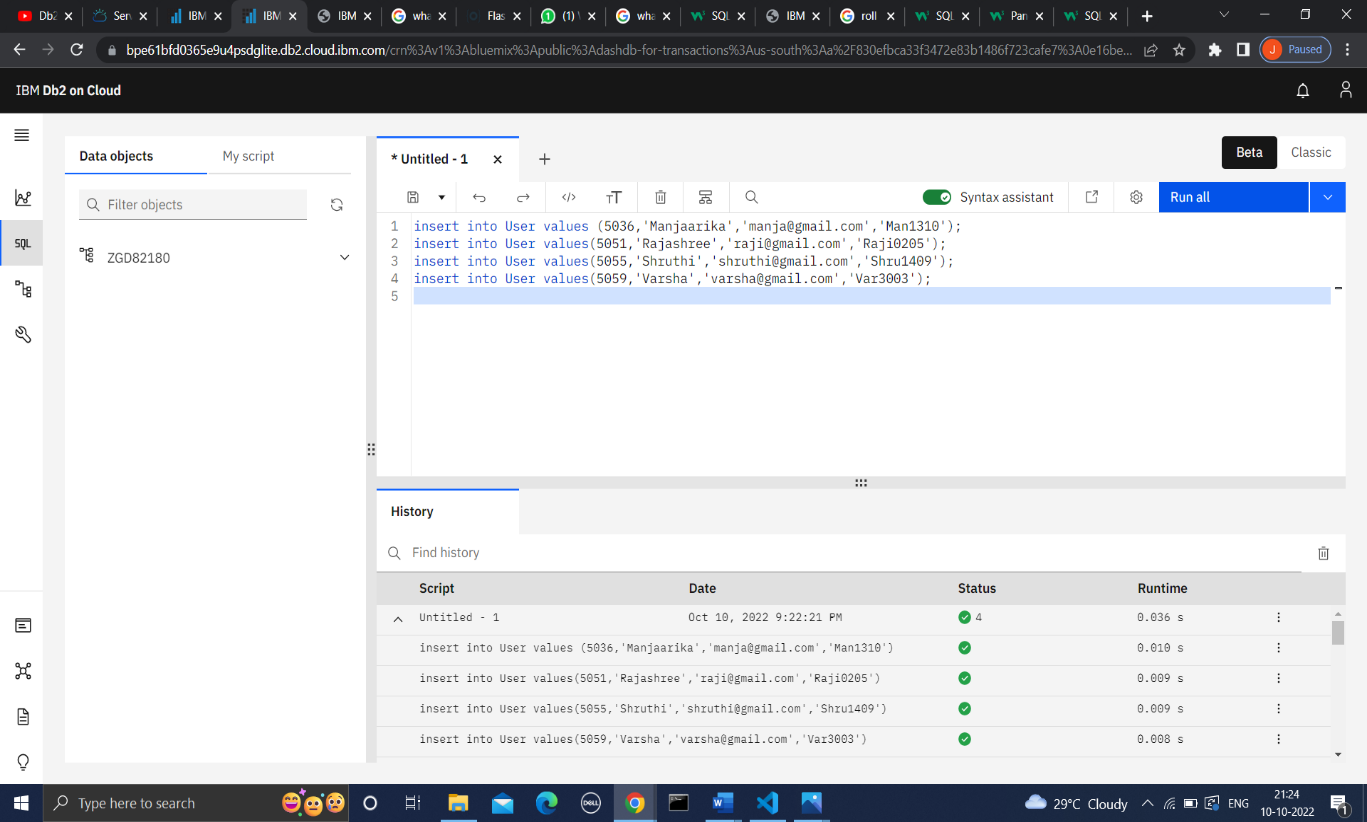
|  |  |
| --- | --- |
| Assignment Date | 26 September 2022 |
| Team ID | PNT2022TMID28027 |
| Project Name | Personal Expense Tracker Application |
| Maximum Marks | 2 Marks |

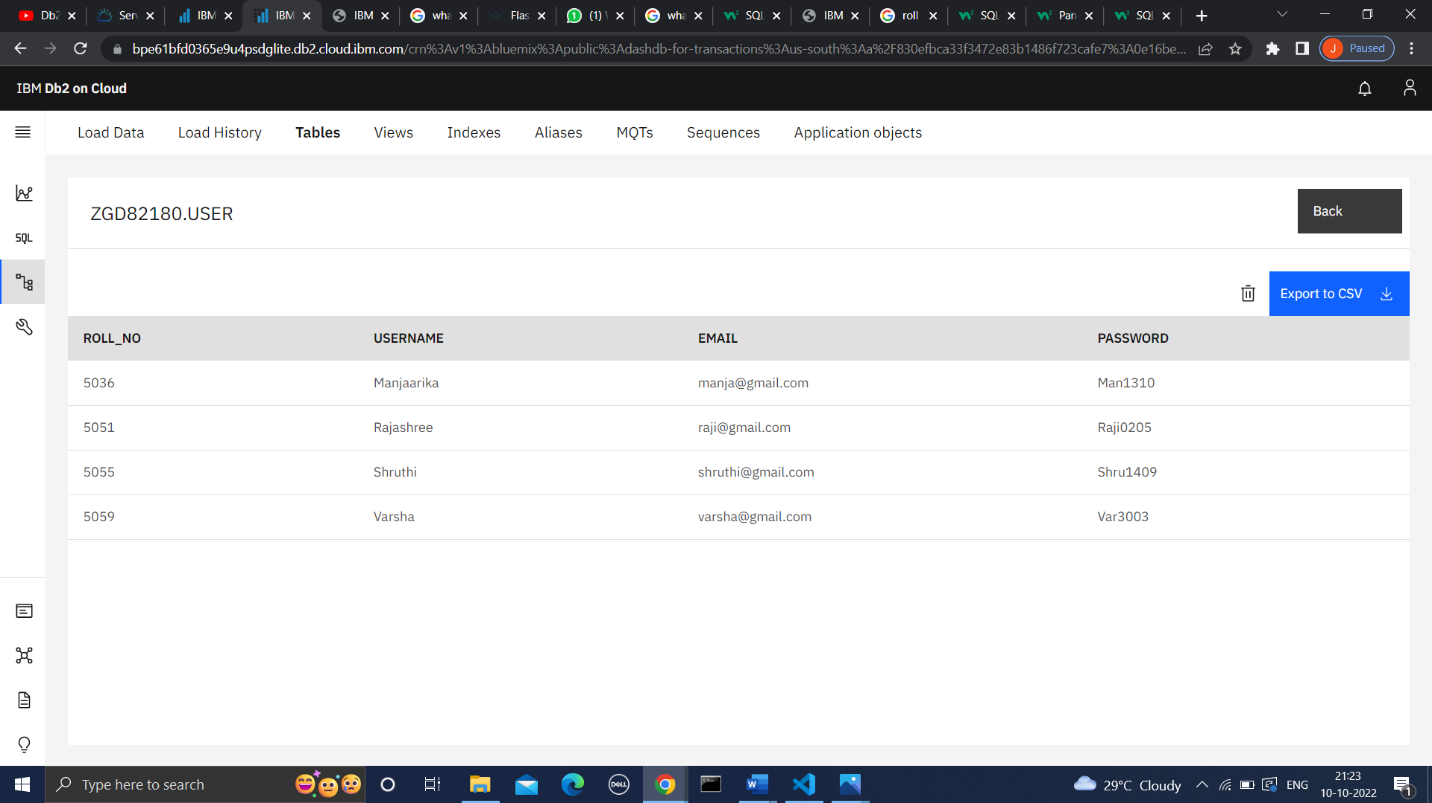
**Question-1:**

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



Insert value into table

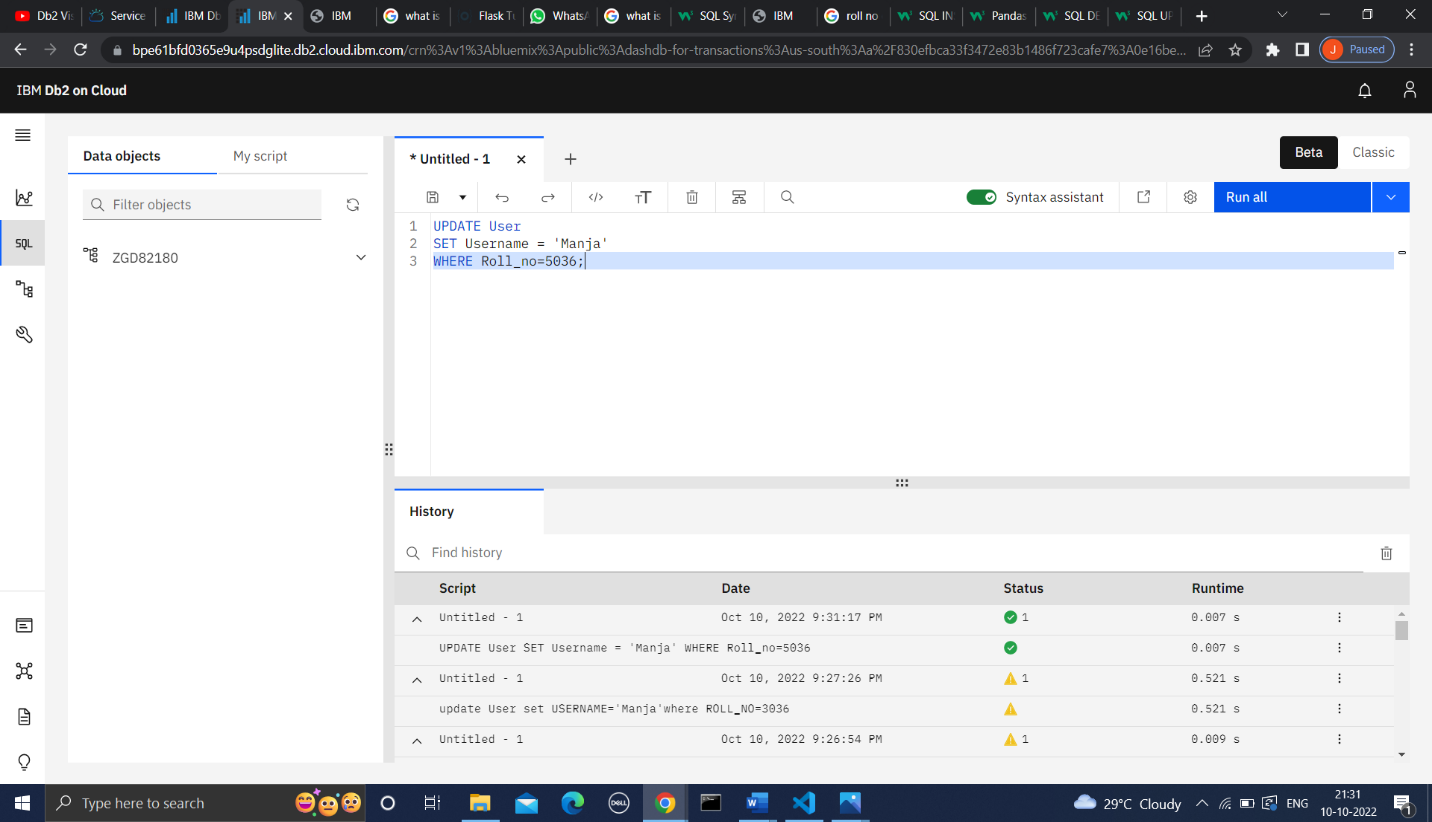


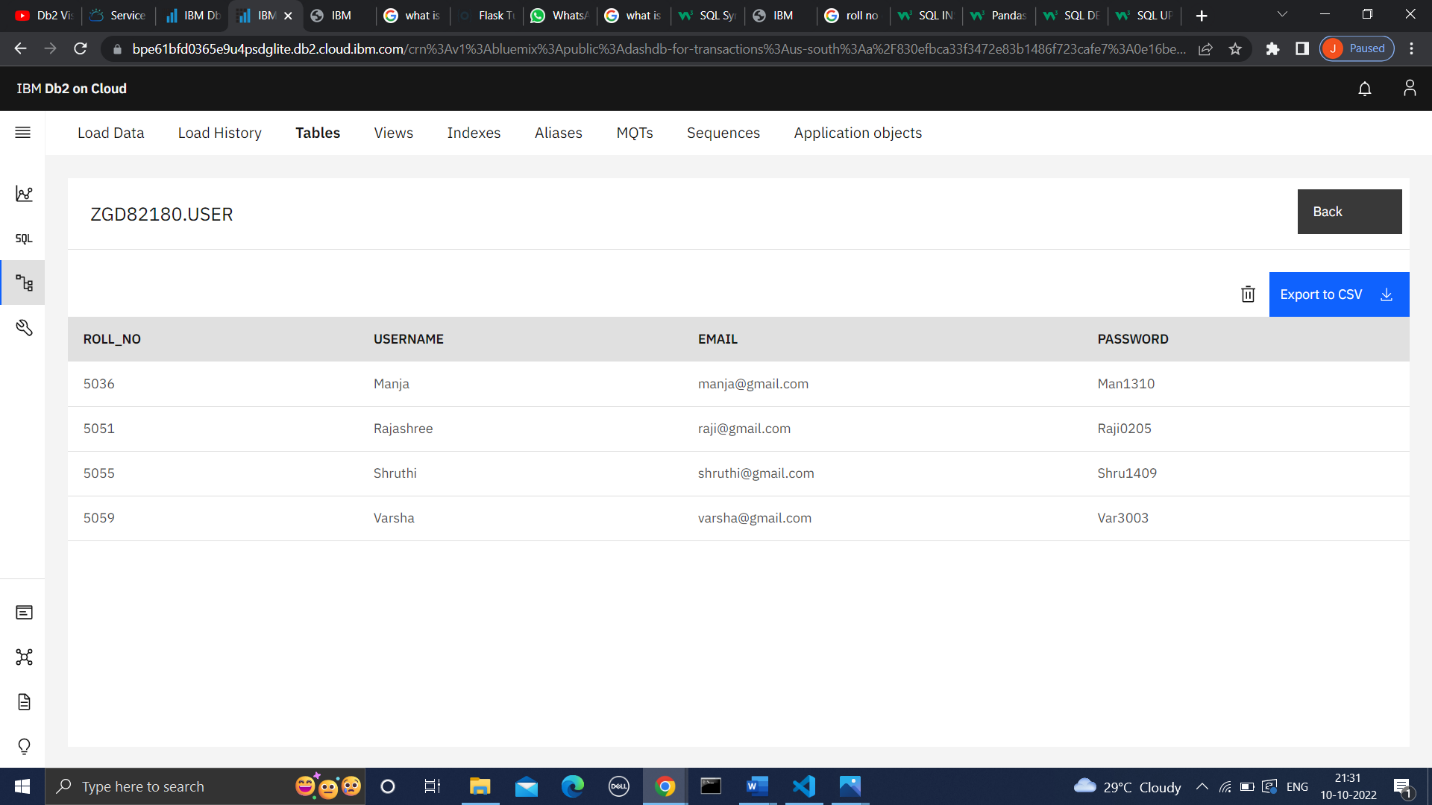


**Question-2:**

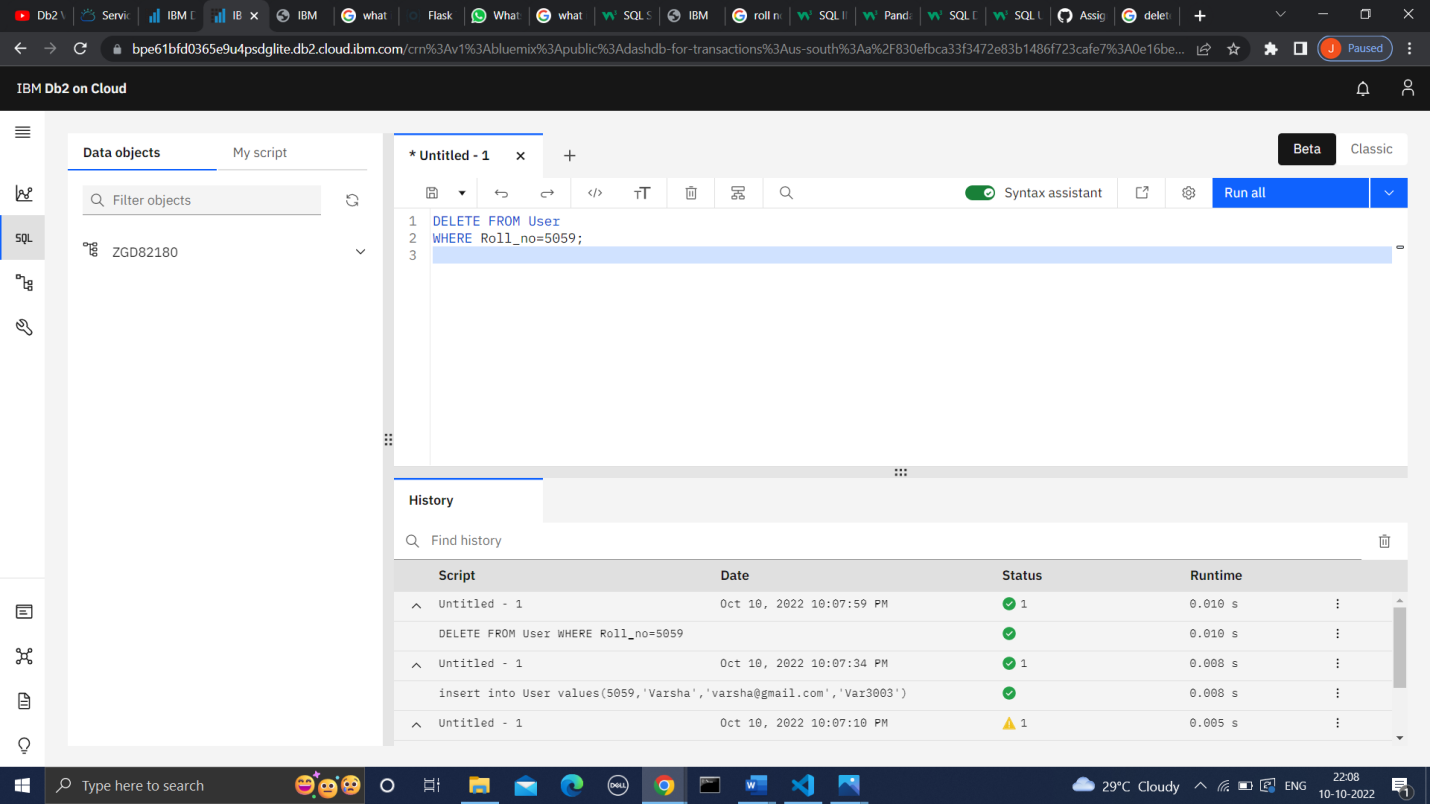
Perform UPDATE, DELETE Queries with user table

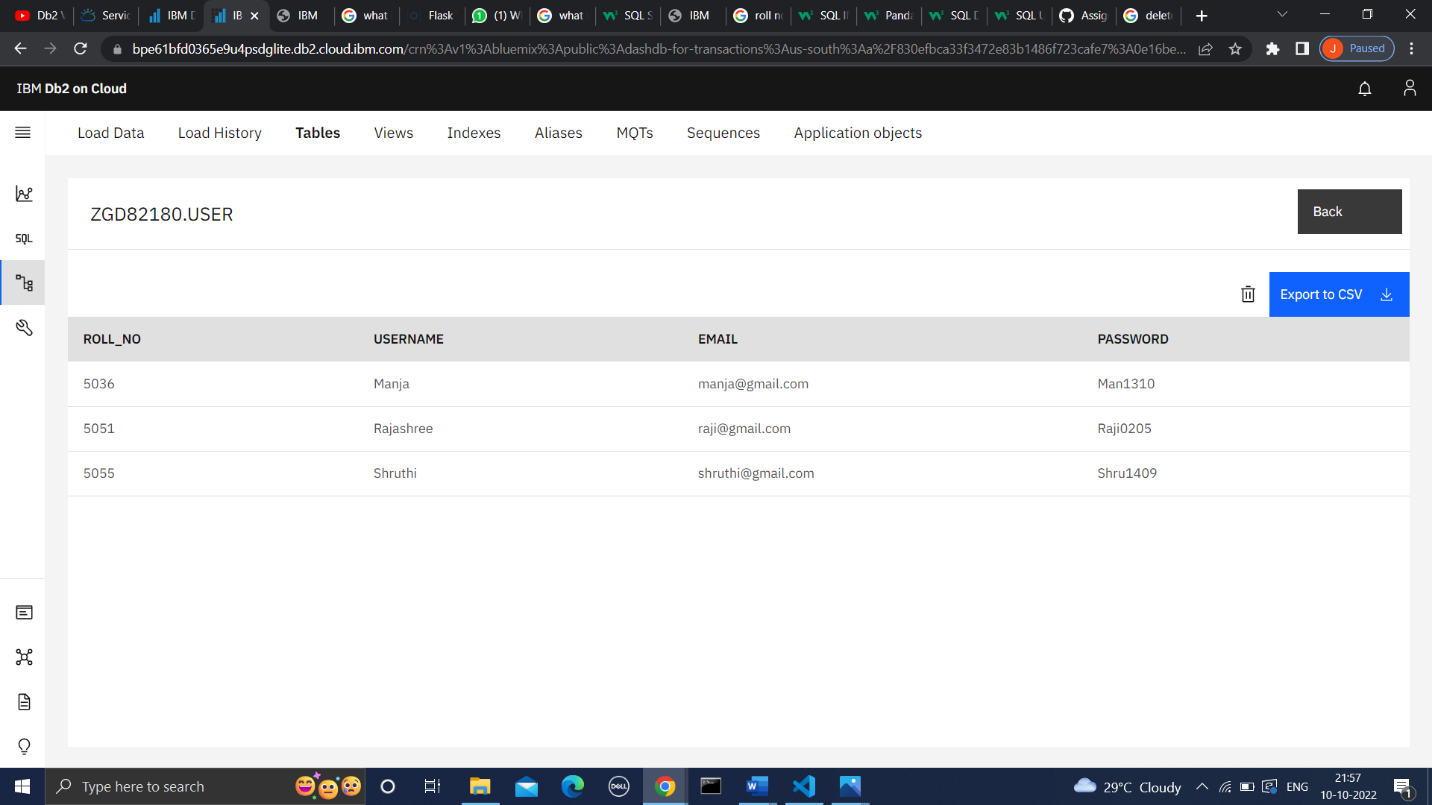
Update the table





Delete a row in the table





**Question 3:**

Connect python code to db2

**Program:**

from flask import Flask

import ibm\_db

app=Flask(\_\_name\_\_)

app.secret\_key='a'

try:

conn=ibm\_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=mld08411;PWD=mCn9EnGnLGGbg2JH","","")

except:

print("Unable to connect: ",ibm\_db.conn\_error())

@app.route("/")

def dashboard():

return “welcome”

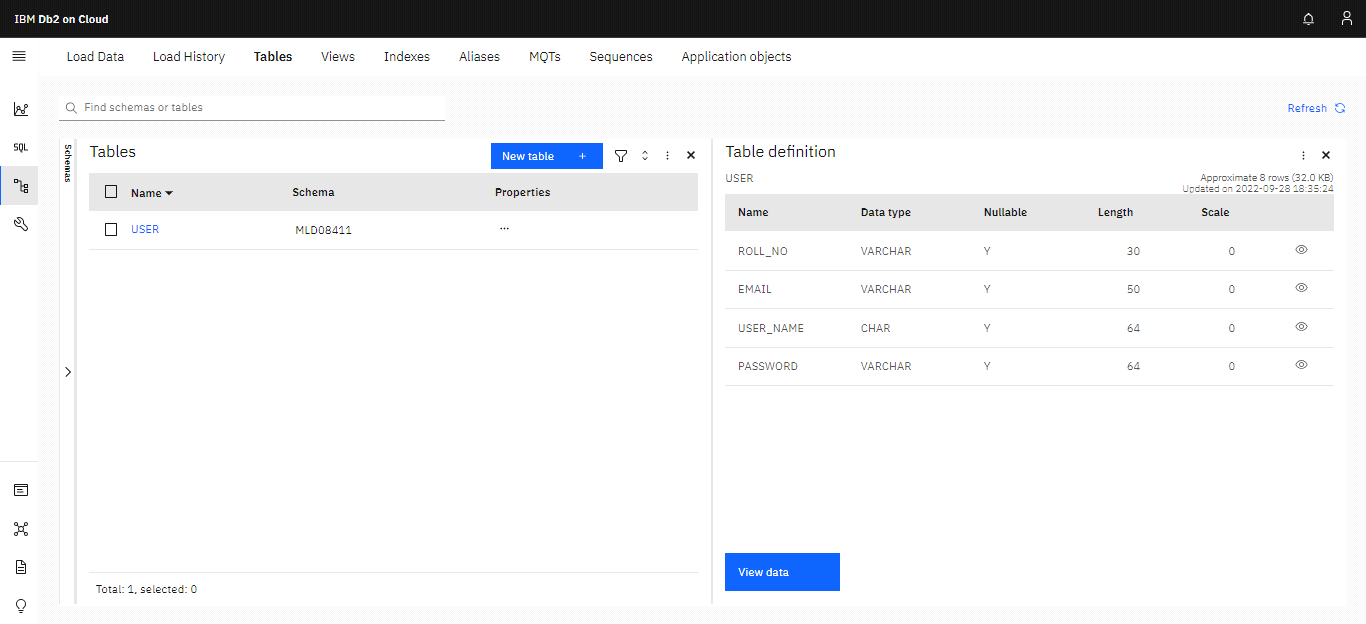
if \_\_name\_\_=='\_\_main\_\_':

app.run(debug=True)

**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 username and password. If the user is valid show the welcome page.

**Database:**



**Program:**

**User.py**

from flask import Flask,flash,render\_template,request,redirect,url\_for,session

import ibm\_db

app=Flask(\_\_name\_\_)

app.secret\_key='a'

try:

conn=ibm\_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=mld08411;PWD=mCn9EnGnLGGbg2JH","","")

except:

print("Unable to connect: ",ibm\_db.conn\_error())

@app.route("/")

def dash():

return render\_template('register.html',msg=" ")

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

def register():

error = None

if request.method=='POST':

username=request.form['username']

email=request.form['email']

roll\_number=request.form['rollnumber']

password=request.form['password']

sql="SELECT \* FROM user WHERE roll\_no=?"

prep\_stmt=ibm\_db.prepare(conn,sql)

ibm\_db.bind\_param(prep\_stmt,1,roll\_number)

ibm\_db.execute(prep\_stmt)

account=ibm\_db.fetch\_assoc(prep\_stmt)

print(account)

if account:

error="Account already exists! Log in to continue !"

else:

insert\_sql="INSERT INTO user values(?,?,?,?)"

prep\_stmt=ibm\_db.prepare(conn,insert\_sql)

ibm\_db.bind\_param(prep\_stmt,1,roll\_number)

ibm\_db.bind\_param(prep\_stmt,2,email)

ibm\_db.bind\_param(prep\_stmt,3,username)

ibm\_db.bind\_param(prep\_stmt,4,password)

ibm\_db.execute(prep\_stmt)

flash(" Registration successfull. Log in to continue !")

else:

pass

return render\_template('login.html',error=error)

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

def login():

error = None

if request.method=='POST':

username=request.form['username']

password=request.form['password']

sql="SELECT \* FROM user WHERE user\_name=? 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['USER\_NAME']

session["username"]=account["USER\_NAME"]

flash("Logged in successfully!")

return redirect(url\_for("welcome\_page"))

else:

error="Incorrect username / password"

return render\_template('login.html',error=error)

@app.route('/welcome')

def welcome\_page():

return render\_template("welcome.html",user=session['id'])

if \_\_name\_\_=='\_\_main\_\_':

app.run(debug=True)

**register.html**

<html>

<head>

<title>Registration page</title>

<link rel="stylesheet" href="{{url\_for('static', filename='style.css')}}">

</head>

<body>

<div class="center"> <br /><br />

<form action="/register" method="POST">

<h2>Student Registration Portal</h2> <br /><br />

<p> Enter Email ID:</p>

<input type="email" name="email"/>

<p> Enter Username:</p>

<input type="text" name= "username" />

<p> Enter Password:</p>

<input type="password" name="password"/>

<p> Enter Roll number:</p>

<input type="text" name="rollnumber"/>

</br></br></br>

<input type="submit" value="submit" />

</form>

</div>

</body>

</html>

**login.html**

<html>

<head>

<title>Login page</title>

<link rel="stylesheet" href="{{url\_for('static', filename='style.css')}}">

</head>

<body>

{% if error %}

<p><strong style="color:red">Error</strong>: {{error}}</p>

{% endif %}

{% with messages = get\_flashed\_messages() %}

{% if messages %}

{% for message in messages %}

<p style="color:green">{{ message }}</p>

{% endfor %}

{% endif %}

{% endwith %}

<div class="center"> <br /><br />

<form action="/login" method="POST">

<h1>Login</h1> <br /><br />

<p> Enter Username:</p>

<input type="text" name= "username" />

<p> Enter Password:</p>

<input type="password" name="password"/>

</br></br></br>

<input type="submit" value="submit" />

</form>

</div>

</body>

</html>

**welcome.html**

<html>

<head>

<title>Login page</title>

<link rel="stylesheet" href="{{url\_for('static', filename='style.css')}}">

</head>

<body>

{% with messages = get\_flashed\_messages() %}

{% if messages %}

{% for message in messages %}

<p style="color:green">{{ message }}</p>

{% endfor %}

{% endif %}

{% endwith %}

<br /><br />

<center>

<h2 > Welcome {{user}}</h2>

</center>

</body>

</html>

**style.css**

.center{

margin: auto;

width: 15%;

}

h2{

text-align:center;

}

input{

border-radius:5px;

padding:7px;

}

**Output:**

