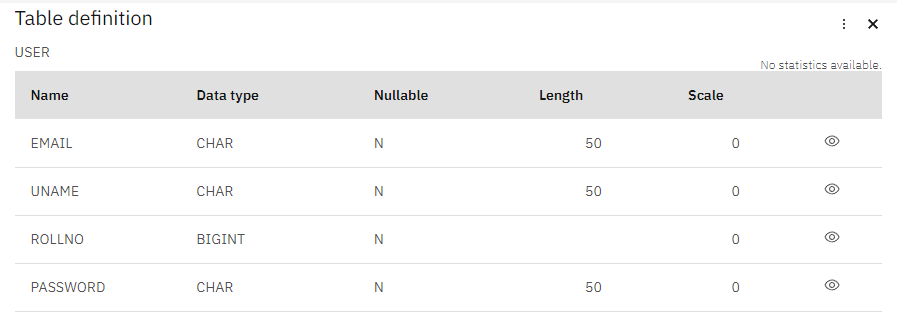
|  |  |
| --- | --- |
| Date | 20 October 2022 |
| Name | Pakanati Hemashika |
| Roll no | 110719104038 |
| Batch | B1-1M3E |

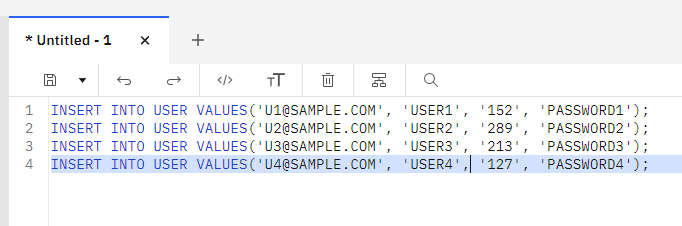
Assignment 2

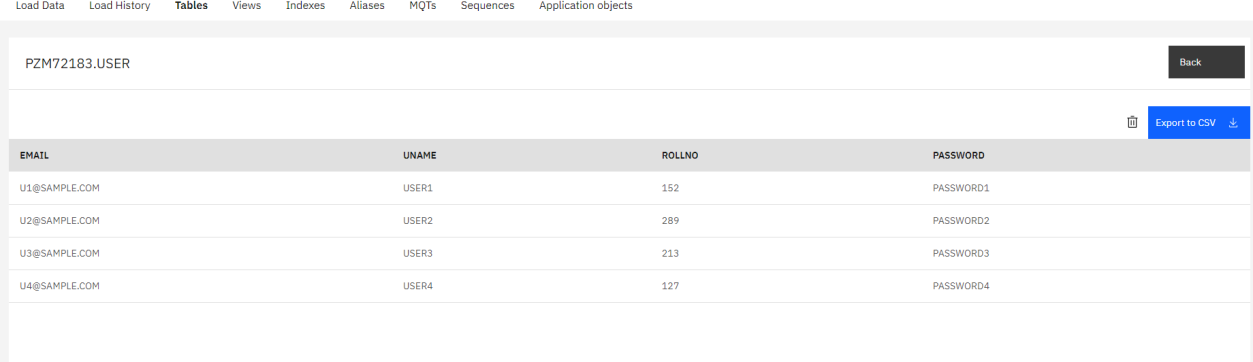
1. *Create User table with user with email, username, roll number, password.*
2. *Perform UPDATE, DELETE Queries with user table*
3. *Connect python code to db2.*
4. *Create a flask app with registration page, login page and welcome page. By default, load the registration page once the user enters all 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*

**Creation of Table**



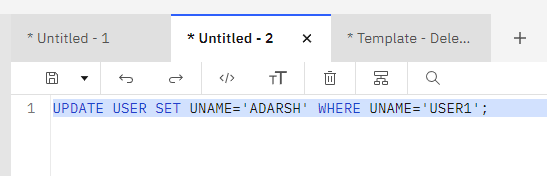
**Insertion of Data**





**Performing Queries on the table**

Update Query



Delete Query

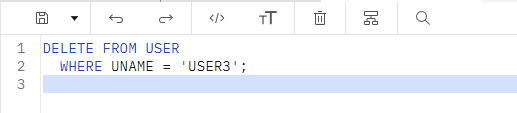
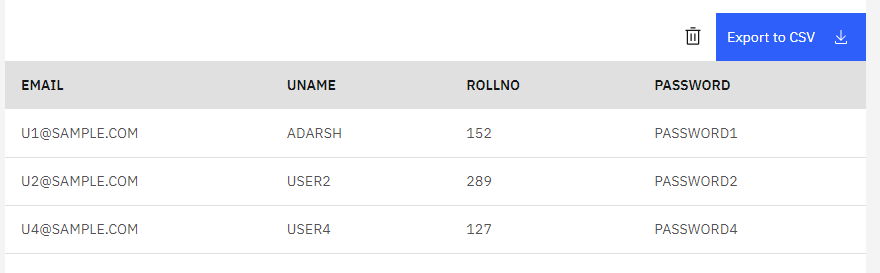


Table after updates



**Connect python code to IBM Db2**

conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0- d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;Securit y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pzm72183;PWD=esgc1kHRU qTUfn9I;", "", "")

**Flask app to create registration form and insert data into Db2**

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

from wtforms import StringField, PasswordField, SubmitField from wtforms.validators import DataRequired, Email, EqualTo import ibm\_db

app = Flask( name ) app.config['SECRET\_KEY'] = 'mysecretkey'

conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0- d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;Securit y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pzm72183;PWD=esgc1kHRU qTUfn9I;", "", "")

class RegistrationForm(FlaskForm):

first\_name = StringField('First Name', validators=[DataRequired()]) last\_name = StringField('Last Name', validators=[DataRequired()]) email = StringField('Email', validators=[DataRequired(), Email()]) phone = StringField('Phone', validators=[DataRequired()])

password = PasswordField('Password', validators=[DataRequired()]) submit = SubmitField('Submit')

@app.route('/', methods=['GET', 'POST']) def index():

return render\_template('index.html')

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

form = RegistrationForm()

if form.validate\_on\_submit():

flash(f'Account created for {form.first\_name.data}

{form.last\_name.data}!', 'success')

return render\_template('success.html', form=form)

sql = "INSERT INTO USER (EMAIL, UNAME, ROLLNO, PASSWORD) VALUES (?, ?, ?,

?)"

stmt = ibm\_db.prepare(conn, sql) ibm\_db.bind\_param(stmt, 1, form.email.data)

ibm\_db.bind\_param(stmt, 2, form.first\_name.data)

ibm\_db.bind\_param(stmt, 3, form.phone.data)

ibm\_db.bind\_param(stmt, 4, form.password.data) ibm\_db.execute(stmt)

return render\_template('success.html', form=form) @app.route('/success', methods=['GET', 'POST'])

def success():

return render\_template('success.html', first\_name=request.args.get('first\_name'), last\_name=request.args.get('last\_name'), email=request.args.get('email'), phone=request.args.get('phone'))

if name == ' main ': app.run(debug=True)