

Assignment -2

Python Programming

Assignment Date	23 September 2022
Student Name	Harini K
Student Roll Number	2019115035
Maximum Marks	2 Marks

Question 1:

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

Solution 1:

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, and the 'STUDENTS' table is listed under the 'DNN32176' schema. The 'Table definition' pane on the right shows the following columns:

Name	Data type	Nullable	Length	Scale
EMAIL	VARCHAR	N	32	0
PASSWORD	VARCHAR	N	32	0
NAME	VARCHAR	N	32	0
ROLLNO	DECIMAL	N	15	0

A 'View data' button is visible at the bottom of the table definition pane.

Question 2:

Perform UPDATE, DELETE Queries with user table

Solution 2:

INSERT

```
insert into students values ('harinik@gmail.com','152427','_harini_',2019115035);
```

```
insert into students values ('tosrini11@gmail.com','4853sri','srinidhi11',2019115105);
```

```
insert into students values ('kmona200181@gmail.com','123456','itsmona14',2019115055);
```

```
insert into students values ('sudhir@gmail.com','762736','sudhir',2019115110);
```

*** INSERT**

```

1 insert into students values ('harinik@gmail.com','152427','_harini_',2019115035);
2 insert into students values ('tosrini11@gmail.com','4853sri','srinidhi11',2019115105);
3 insert into students values ('kmona200181@gmail.com','123456','itsmona14',2019115055);
4 insert into students values ('sudhir@gmail.com','762736','sudhir',2019115110);

```

History

Script	Date	Status	Runtime
INSERT	Nov 6, 2022 7:42:13 PM	✓ 4	0.024 s
insert into students values ('harinik@gmail.com','1524...		✓	0.008 s
insert into students values ('tosrini11@gmail.com','48...		✓	0.005 s
insert into students values ('kmona200181@gmail.com','...		✓	0.005 s
insert into students values ('sudhir@gmail.com','76273...		✓	0.006 s

Output:

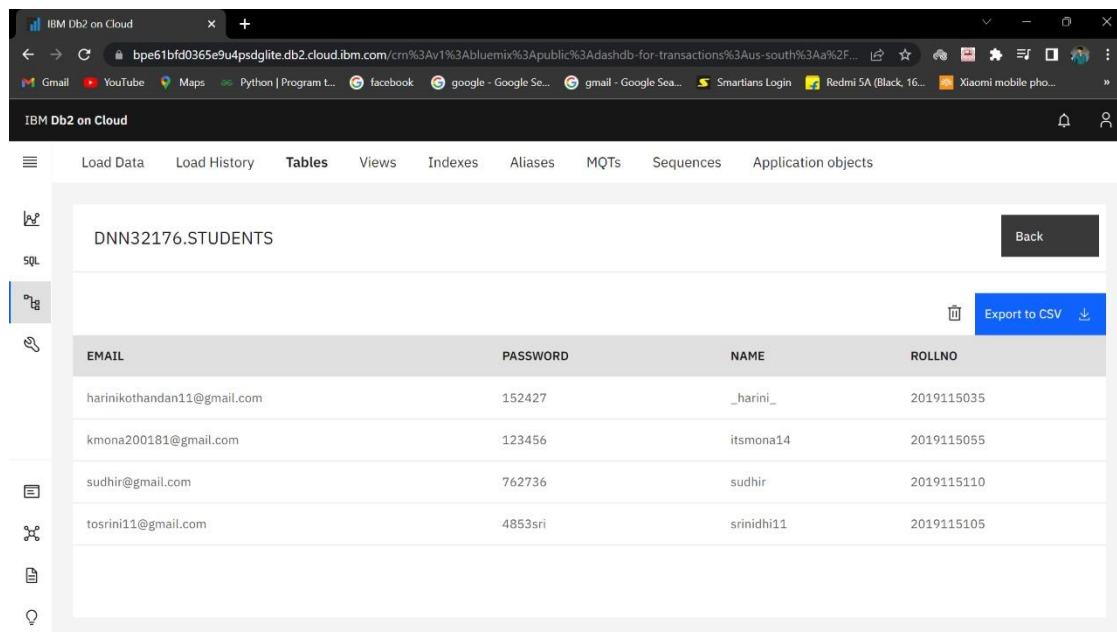
DNN32176.STUDENTS

EMAIL	PASSWORD	NAME	ROLLNO
harinik@gmail.com	152427	_harini_	2019115035
kmona200181@gmail.com	123456	itsmona14	2019115055
sudhir@gmail.com	762736	sudhir	2019115110
tosrini11@gmail.com	4853sri	srinidhi11	2019115105

UPDATE

update students set email='harinikothandan11@gmail.com' where rollno=2019115035;

Output:

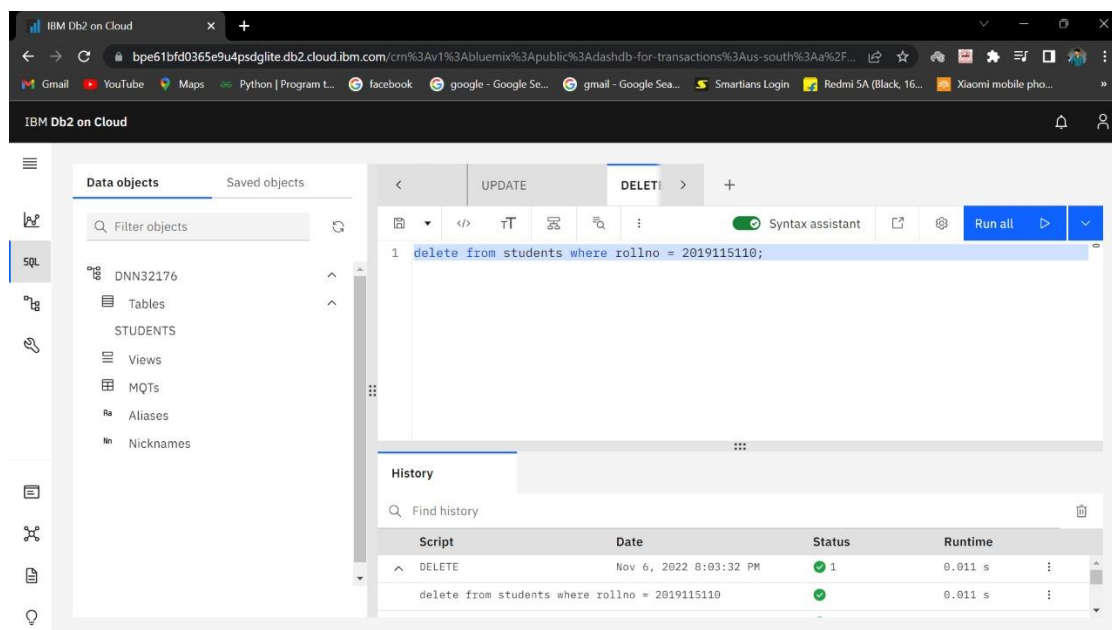


The screenshot shows the IBM Db2 on Cloud web interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected, displaying a table named 'DNN32176.STUDENTS'. A 'Back' button is in the top right. Below the table name, there are icons for deleting the table and an 'Export to CSV' button. The table itself has four columns: 'EMAIL', 'PASSWORD', 'NAME', and 'ROLLNO'. It contains five rows of data.

EMAIL	PASSWORD	NAME	ROLLNO
harinikothandan11@gmail.com	152427	_harini_	2019115035
kmona200181@gmail.com	123456	itsmona14	2019115055
sudhir@gmail.com	762736	sudhir	2019115110
tosrini11@gmail.com	4853sri	srinidhi11	2019115105

DELETE

delete from students where rollno = 2019115110;



The screenshot shows the IBM Db2 on Cloud web interface with the SQL editor open. The 'Data objects' panel on the left shows the database structure. The SQL editor has a 'DELETE' tab selected, and the SQL statement 'delete from students where rollno = 2019115110;' is entered. Below the editor, the 'History' panel shows the execution of the DELETE statement, which was successful.

Script	Date	Status	Runtime
DELETE	Nov 6, 2022 8:03:32 PM	✓ 1	0.011 s
delete from students where rollno = 2019115110		✓	0.011 s

EMAIL	PASSWORD	NAME	ROLLNO
harinikothandan11@gmail.com	152427	_harini_	2019115035
kmoma200181@gmail.com	123456	itsmona14	2019115055
tosrini11@gmail.com	4853sri	srinidhi11	2019115105

Question 3:

Connect python code to db2.

Solution 3:

```
# -*- coding: utf-8 -*-
```

```
''''''
```

```
@author: Harini
```

```
''''''
```

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

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

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=125f9f61-9715-46f9-9399-
c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30426;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dnn32176;PWD=sSCCeOSMo3JZxAef",",")
```

```
if(conn):
    print("CONNECTED SUCCESSFULLY")
    print("Connection : "+str(conn))
```

```
sql="SELECT * FROM students WHERE rollno=2019115035"
email="harinikothandan11@gmail.com"
stmt = ibm_db.prepare(conn,sql)
ibm_db.execute(stmt)
acc = ibm_db.fetch_assoc(stmt)
```

```
if acc:
    print(acc)
```

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

```

from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3

if(conn):
    print("CONNECTED SUCCESSFULLY")
    print("Connection : "+str(conn))
    sql="SELECT * FROM USER WHERE rollno=2019115055"
    email="kmona200181@gmail.com"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.execute(stmt)
    acc = ibm_db.fetch_assoc(stmt)
    if acc:
        print(acc)

if __name__ == '__main__':
    app.run()

```

Output:

```

In [12]: runfile('E:/Study materials/Sem 7/IBM/Exercise/Assignment2/temp.py', wdir='E:/
Study materials/Sem 7/IBM/Exercise/Assignment2')
CONNECTED SUCCESSFULLY
Connection : <ibm_db.IBM_DBConnection object at 0x000002003B9B6BB0>
{'EMAIL': 'kmona200181@gmail.com', 'PASSWORD': '123456', 'USERNAME': 'itsmona14',
'ROLLNO': '2019115055'}
* Serving Flask app "temp" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

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

Solution 4:

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
@author: Harini
```

```
"""
```

```
from flask import Flask, render_template, request, redirect, url_for, 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=lfx34122;PWD=jmQDS9wCaxqRIIQd", "", "")
```

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

```
def login():
```

```
    global userid
```

```
    msg = "
```

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

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

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

```
        sql = "SELECT * FROM USERS WHERE EMAIL=? AND PASSWORD=?"
```

```
        stmt = ibm_db.prepare(conn, sql)
```

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

```
        ibm_db.bind_param(stmt, 2, password)
```

```
        ibm_db.execute(stmt)
```

```

acc = ibm_db.fetch_assoc(stmt)

print(acc)

if acc:

    session['loggedin'] = True

    session['id'] = acc['USERNAME']

    userid = acc['USERNAME']

    session['username'] = acc['USERNAME']

    msg = acc['USERNAME']


    return render_template('dashboard.html', msg = msg)

else:

    msg = "Incorrect username/password!!"

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

```

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

```
def register():
```

```

    msg = ""

    if request.method == 'POST':

        username = request.form['username']

        email = request.form['email']

        password = request.form['password']

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

        stmt = ibm_db.prepare(conn,sql)

        ibm_db.bind_param(stmt,1,username)

        ibm_db.execute(stmt)

        acc = ibm_db.fetch_assoc(stmt)

        print(acc)

```

```

if acc:

    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 = "Name must contain only characters and numbers !!"

else:

    sql = "INSERT INTO USERS VALUES (?, ?, ?)"

    stmt = ibm_db.prepare(conn,sql)

    ibm_db.bind_param(stmt,1,username)

    ibm_db.bind_param(stmt,2,email)

    ibm_db.bind_param(stmt,3,password)

    ibm_db.execute(stmt)

    msg = "Successgully registered !!Login to continue"

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

elif request.method == 'POST':

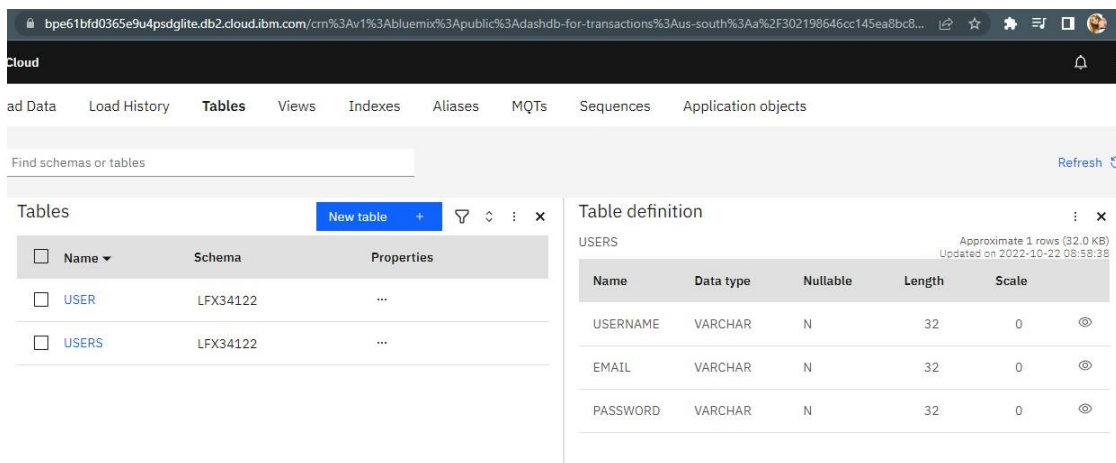
    msg = "Please fill out the form !"

    return render_template('register.html', msg = msg)

if __name__ == '__main__':

    app.run()

```



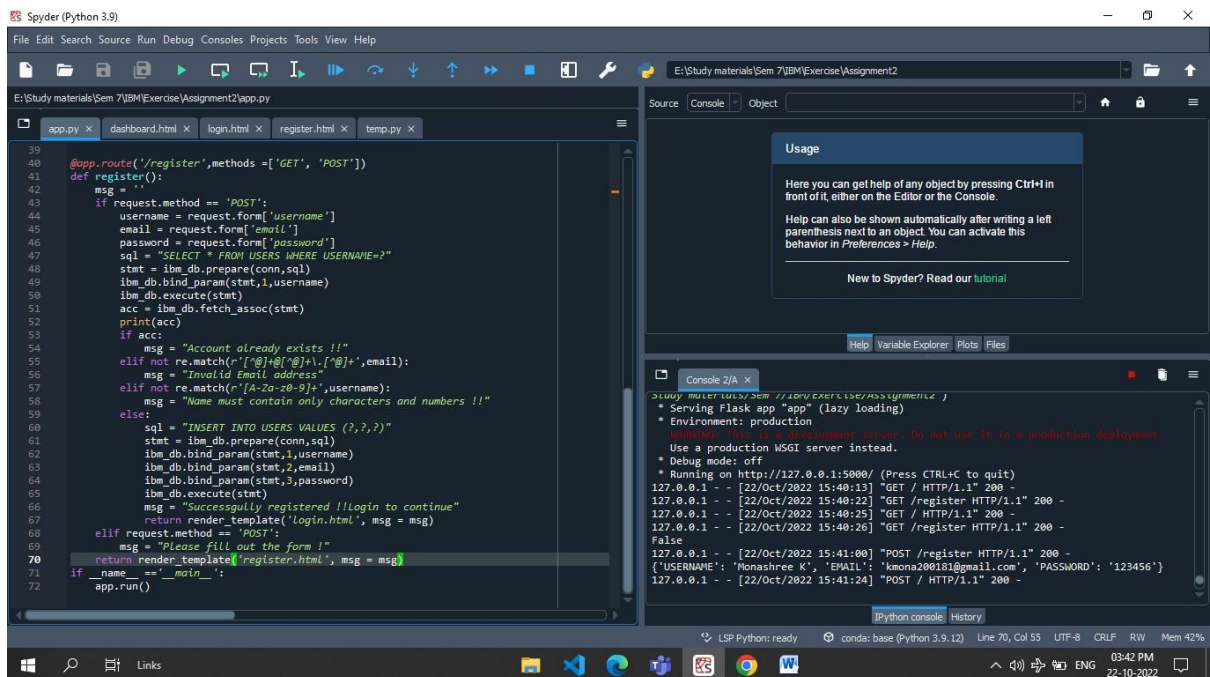
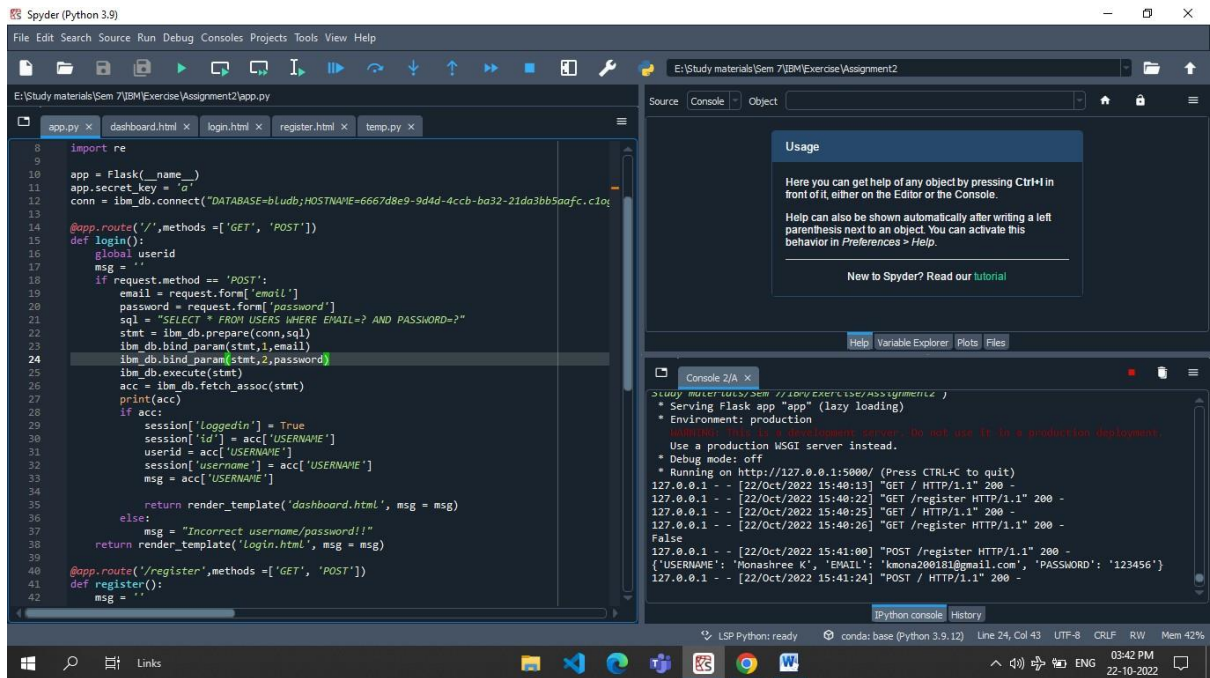
The screenshot shows the IBM Cloud SQL console interface. The top navigation bar includes 'ad Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected, and a search bar for 'Find schemas or tables' is visible. Below the search bar, a table lists the available tables:

Name	Schema	Properties
USER	LPX34122	...
USERS	LPX34122	...

The 'USERS' table is selected, and the 'Table definition' panel on the right displays its structure:

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

Additional information for the 'USERS' table: Approximate 1 rows (32.0 KB), Updated on 2022-10-22 08:58:38.



Output:

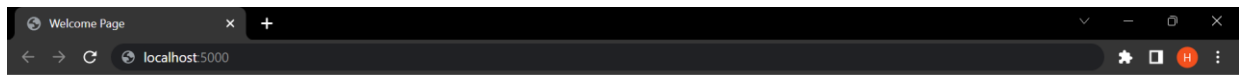
Register page

The screenshot shows a web browser with three tabs: 'Welcome Page', 'Login form | Personal expense Tr...', and 'Sign Up Form'. The address bar shows 'localhost:5000'. The main content area has a light orange background. In the center is a dark blue rounded rectangle with a grey drop shadow. Inside this rectangle, the text 'SIGN UP' is at the top in white, with 'Personal Expense Tracker' below it in a smaller white font. There are three input fields: the first contains 'Harini K', the second contains 'harinikothandan11@gmail.com', and the third contains masked characters '*****'. Below the input fields is a blue rounded button with the text 'SIGN UP'. At the bottom of the rectangle, it says 'Already have an account? [Log In](#)' in white text.

Login page

The screenshot shows the same web browser setup as the previous image. The main content area has a light orange background. In the center is a white rounded rectangle with a pink drop shadow. Inside this rectangle, the text 'LOG IN' is at the top in black, with 'Personal Expense Tracker' below it in a smaller black font. There are two input fields: the first contains 'harinikothandan11@gmail.com' and the second contains masked characters '*****'. Below the input fields is a green rounded button with the text 'Let's start!'. At the bottom of the rectangle, it says 'Don't have an account? [Sign up](#)' in black text.

Welcome page



Welcome Harini K.....

Table

A screenshot of the IBM Db2 on Cloud console. The 'Tables' tab is selected. The table 'DNN32176.USERS' is displayed. The table has three columns: USERNAME, EMAIL, and PASSWORD. The data row shows 'Harini K', 'harinikothandan11@gmail.com', and 'Harini1!'. There is an 'Export to CSV' button and a 'Back' button.

USERNAME	EMAIL	PASSWORD
Harini K	harinikothandan11@gmail.com	Harini1!