

## Assignment 2

Team ID	PNT2022TMID14117
Project Name	Plasma Donor Application

### Question:

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

### Solution:

```
CREATE TABLE USER(  
    USER_ID INT GENERATED BY DEFAULT AS IDENTITY NOT  
    NULL,  
    EMAIL_ID VARCHAR(150) NOT NULL,  
    USER_NAME VARCHAR(255) NOT NULL,  
    ROLL_NUMBER VARCHAR(100) NOT NULL,  
    PASSWORD VARCHAR(100) NOT NULL,  
    PRIMARY KEY (USER_ID)  
);  
INSERT INTO USER(EMAIL_ID,  
USER_NAME,ROLL_NUMBER,PASSWORD)  
VALUES('rk9166771@gmail.com','Rajeshkumar  
S','19CS123','9787234640');  
  
INSERT INTO USER(EMAIL_ID,  
USER_NAME,ROLL_NUMBER,PASSWORD)  
VALUES('ramachandramoorthykb@gmail.com','Ramachandramoorthy K  
B','19CS124','9912346578');
```

The screenshot shows the IBM Db2 SQL Editor interface. The left pane contains a SQL script with the following content:

```

1 CREATE TABLE USER(
2   USER_ID INT GENERATED BY DEFAULT AS IDENTITY NOT NULL,
3   EMAIL_ID VARCHAR(150) NOT NULL,
4   USER_NAME VARCHAR(255) NOT NULL,
5   ROLL_NUMBER VARCHAR(100) NOT NULL,
6   PASSWORD VARCHAR(100) NOT NULL,
7   PRIMARY KEY (USER_ID)
8 );
9
10 INSERT INTO USER(EMAIL_ID,USER_NAME,ROLL_NUMBER,PASSWORD)
11 VALUES('rk9166771@gmail.com','Rajeshkumar S','19CS123','9787234640');
12
13 INSERT INTO USER(EMAIL_ID, USER_NAME,ROLL_NUMBER,PASSWORD)
14 VALUES('ramachandramoorthykb@gmail.com','Ramachandramoorthy K B','19CS124','991
15
16 SELECT * FROM USER;
17
18 UPDATE USER SET EMAIL_ID='dev.rajeshkumars@gmail.com' WHERE USER_NAME='Rajeshku
19
20 SELECT * FROM USER;
21
22 DELETE FROM USER WHERE USER_NAME = 'Rajeshkumar S';
23
24 SELECT * FROM USER;
25
26 DROP TABLE USER;
27

```

The right pane shows the execution results for the query `SELECT * FROM USER`. The run time is 0.008 s. The results are displayed in a table with the following columns: `USER_ID`, `EMAIL_ID`, `USER_NAME`, and `ROLL_NU`.

USER_ID	EMAIL_ID	USER_NAME	ROLL_NU
1	rk9166771@gmail.com	Rajeshkumar S	19CS123
2	ramachandramoorthykb@gmail.com	Ramachandramoorthy K B	19CS124

2.Perform UPDATE, DELETE Queries with user table.

Solution:

UPDATE:

UPDATE USER SET EMAIL\_ID='dev.rajeshkumars@gmail.com' WHERE USER\_NAME='Rajeshkumar S';

The screenshot shows the IBM Db2 SQL Editor interface after executing the UPDATE query. The left pane contains the same SQL script as before, but the right pane shows the execution results for the query `SELECT * FROM USER`. The run time is 0.016 s. The results are displayed in a table with the following columns: `USER_ID`, `EMAIL_ID`, `USER_NAME`, and `ROLL_NU`.

USER_ID	EMAIL_ID	USER_NAME	ROLL_NU
1	dev.rajeshkumars@gmail.com	Rajeshkumar S	19CS123
2	ramachandramoorthykb@gmail.com	Ramachandramoorthy K B	19CS124

DELETE:

## DELETE FROM USER WHERE USER\_NAME = 'Rajeshkumar S';

The screenshot shows the IBM Db2 Cloud console interface. On the left, a SQL script is being executed. The script includes a CREATE TABLE statement for 'USER', followed by two INSERT statements, an UPDATE statement, and a DELETE statement targeting 'Rajeshkumar S'. The 'DELETE' statement is highlighted. On the right, the execution results are displayed. The 'Result set 1' shows a table with four columns: USER\_ID, EMAIL\_ID, USER\_NAME, and ROLL\_NO. The table contains one row with the following values: 2, ramachandramoorthykb@gmail.com, Ramachandramoorthy K B, and 19CS124. The run time for the query is 0.003 s.

USER_ID	EMAIL_ID	USER_NAME	ROLL_NO
2	ramachandramoorthykb@gmail.com	Ramachandramoorthy K B	19CS124

3. Connect python code to db2.

Solution:

```
def Connection():
```

```
    try:
```

```
        conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=xjk42376;PASSWORD=liEWs4fS57ABi3h1", "", "")
```

```
        print ("Database Connected Successfully!")
        return conn
    except:
```

```
        print ("Unable to connect: ", ibm_db.conn_errormsg())
```

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.

```
def Create(email,name,phone,password,conn):

    columns = "UNAME","UEMAIL","UPHONE","UPASSWORD"
    val = ""+name+"",""+email+"",""+phone+"",""+password+"
    sql = 'Insert into XJK42376.USER(' + columns + ') values('+val+')'
    try:
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.execute(stmt)
        print ("added :-")
        return 1
    except:
        print("Error While Adding the User ! ")
        return 0

def Signin(email,password,conn):

    sql = "SELECT * FROM XJK42376.USER"
    try:
        result = ibm_db.exec_immediate(conn,sql)
        tuple = ibm_db.fetch_tuple(result)
        while tuple != False:
            if str(tuple[1]) == email and str(tuple[3]) == password:
                res = [str(tuple[0]),str(tuple[1]),str(tuple[2])]
                return res
            tuple = ibm_db.fetch_tuple(result)
        print("Fetch Success :-")
        return 0
    except:
        print("fetch not found !")
        return 0
```

Output:

Signup Page:

## Sign Up

**SIGN UP**

---

Already Have an Account ! [Login](#)

Login Page:

Sign In

**SIGN IN**

---

Don't Have an Account ! [Sign Up](#)

When Email or password is Invalid:

Sign In

Invalid Email or Password

Email

Password

SIGN IN

Don't Have an Account ! [Sign Up](#)

Table:

XJK42376.USER

Back

Export to CSV

UNAME	UEMAIL	UPHONE	UPASSWORD
Rajeshkumar	rk9166771@gmail.com	9787234640	123456789
Ramachandramoorthy K B	ramchandramoorthykb@gmail.com	1029384756	123456
Ramya s	ramya@gmail.com	123456788	1234567890
Umapathi K	umapathik@gmail.com	8765432211	1234567890

Home Page:

# DashBoard

Welcome Rajeshkumar !