

Assignment 2

Team ID	PNT2022TMID13787
Project Name	Customer Care Registry

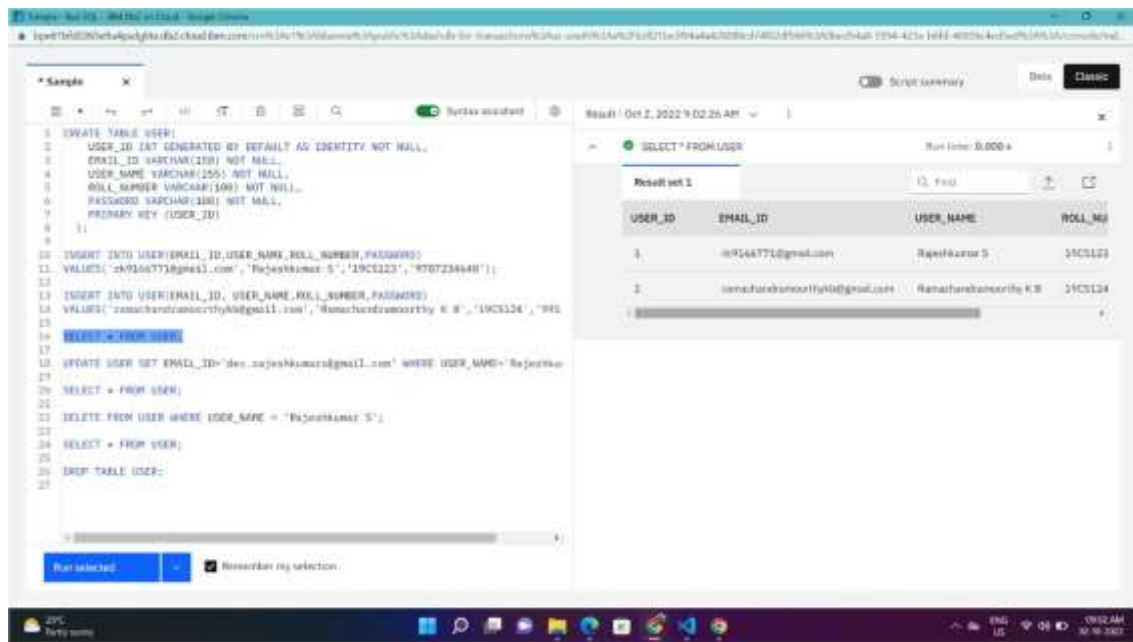
Question:

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');
```

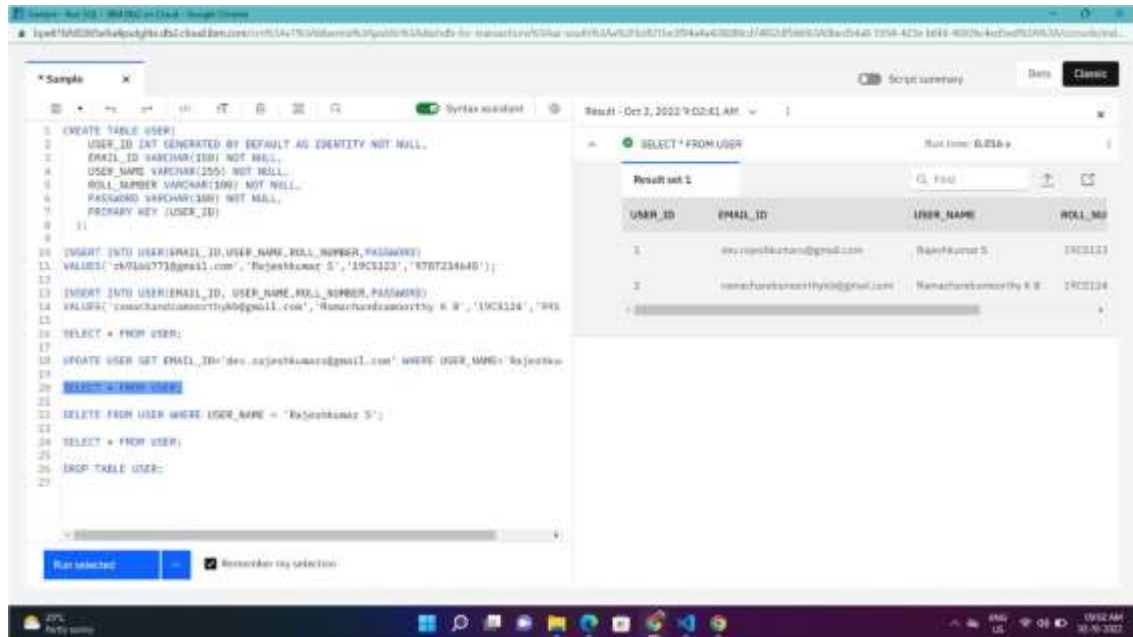


2. Perform UPDATE, DELETE Queries with user table.

Solution:

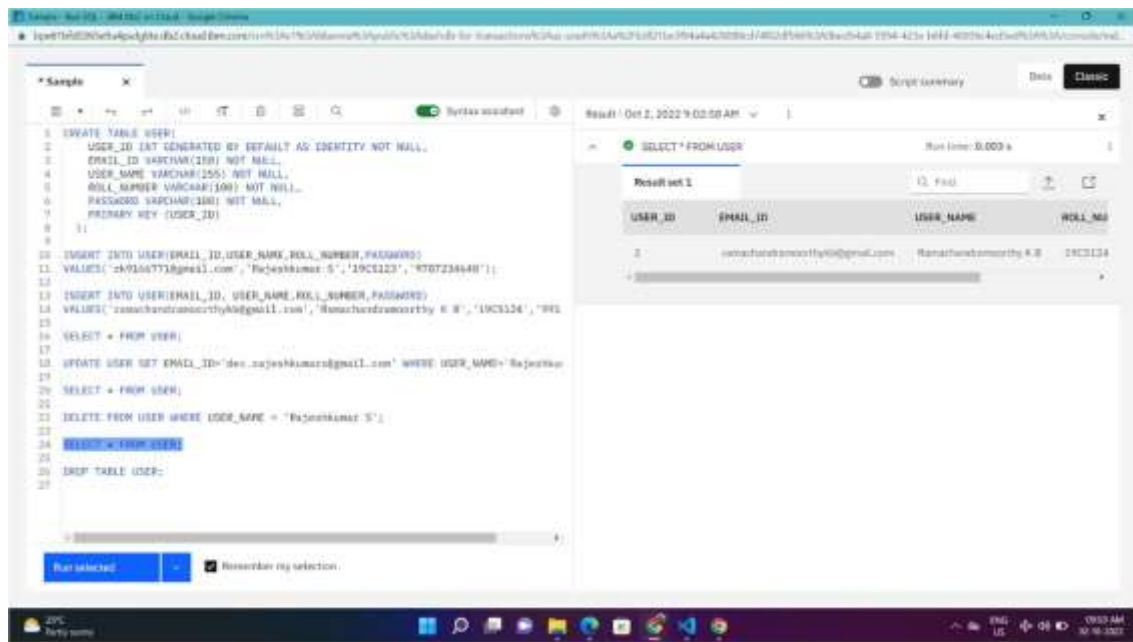
UPDATE:

UPDATE USER SET EMAIL_ID='dev.rajeshkumars@gmail.com' WHERE USER_NAME='Rajeshkumar S';



DELETE:

DELETE FROM USER WHERE USER_NAME = 'Rajeshkumar S';



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;PWD=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

Email

Name

Phone

Password

SIGN UP

Already Have an Account ! [Login](#)

Login Page:

Sign In

Email

Password

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	rk9165773@gmail.com	9787234648	123456789
Ramchandrasekar R B	ramchandrasekarfb@gmail.com	1029384756	123456
Ramya S	ramyas@gmail.com	123456789	(123456789)
Umaapathi K	umapathik@gmail.com	876543211	1234567890

Home Page:

DashBoard

Welcome Rajeshkumar !