

## UPDATE TABLE:

IBM Db2 on Cloud

Data objects

XQR92747

\*Untitled - 1

```
1 INSERT INTO SIGNUP values('YUVARAJ','60','yuvaraj.it19@mamcet.com','Yuvaraj@!@#');
```

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Nov 7, 2022 8:14:26 PM	✓ 1	0.005 s
INSERT INTO SIGNUP values('YUVARAJ','60','yuvaraj.it19@mamcet.com','Yuvaraj@...		✓	0.005 s

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

XQR92747.SIGNUP

Back

Export to CSV

NAME	Roll No	EMAIL	PASSWORD
YUVARAJ	60	yuvaraj.it19@mamcet.com	Yuvaraj@!@#

IBM Db2 on Cloud

Data objects

SQL

XQR92747

\*Untitled - 1

```
1 INSERT INTO SIGNUP values('YAKESH','30','yakesh.it19@mamcet.com','Yakesh@#%');
```

History

Find history

Script	Date	Status	Runtime
^ Untitled - 1	Nov 7, 2022 8:16:55 PM	✓ 1	0.007 s
INSERT INTO SIGNUP values('YAKESH','30','yakesh.it19@mamcet.com','Yakesh@#%...		✓	0.007 s
^ Untitled - 1	Nov 7, 2022 8:14:26 PM	✓ 1	0.005 s
INSERT INTO SIGNUP values('YUVARAJ','60','yuvaraj.it19@mamcet.com','Yuvaraj@...		✓	0.005 s

DELETE:

IBM Db2 on Cloud

Data objects

SQL

XQR92747

\*Untitled - 1

```
1 DELETE FROM SIGNUP WHERE NAME = 'YUVARAJ'
```

History

Find history

Script	Date	Status	Runtime
^ Untitled - 1	Nov 7, 2022 8:20:01 PM	✓ 2	0.013 s
INSERT INTO SIGNUP values('YAKESH','30','yakesh.it19@mamcet.com','Yakesh@#%...		✓	0.007 s
DELETE FROM SIGNUP WHERE NAME = 'YUVARAJ'		✓	0.006 s
^ Untitled - 1	Nov 7, 2022 8:16:55 PM	✓ 1	0.007 s
INSERT INTO SIGNUP values('YAKESH','30','yakesh.it19@mamcet.com','Yakesh@#%...		✓	0.007 s

AFTER DELETION:

[illegible]

## PYTHON DATABASE CONNECTIVITY:

```
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=Db2New;HOSTNAME=19af6446-6171-4641-8aba-
9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;USERNAM
E=kgb92696;PASSWORD=3IlWyGYU3bMIvwnX;SECURITY=SSL;SSLSERVERCERTIFICATE=DigiCer
tGlobalRootCA.crt;", "", "")

@app.route("/", 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)
    account = ibm_db.fetch_assoc(stmt)
print(account)    if account:
    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:
            insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, username)
            ibm_db.bind_param(prepare_stmt, 2, email)
            ibm_db.bind_param(prepare_stmt, 3, password)
            ibm_db.execute(prepare_stmt)
            msg = 'You have successfully registered !'
            elif request.method == 'POST':
                msg = 'Please fill out the form !'
            return render_template('register.html', msg=msg)

@app.route('/login', methods=['GET', 'POST']) def login():
    global
    userid      msg = ''
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE username =? 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['USERID']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
            return render_template('dashboard.html', msg=msg)
        else:
            msg = 'Incorrect username / password !'
            return render_template('login.html', msg=msg)

if __name__ == '__main__':
    app.run(host='0.0.0.0')
    # app.run(debug=True)

```