Databases through Python-Flask and MariaDB

Tanmay Agarwal, Durga Keerthi and G V V Sharma*

Contents

***Abstract*—Databases software applications for small establishments like schools, shops, etc.. can be easily built using the MariaDB database, Python-Flask connector and HTML. This manual shows how to install these free software tools and build a simple application using them.**

1 Python-flask

Flask is Python framework for creating web applications.

1.1 Installation

1) Run the following commands on the terminal

```
sudo apt-get update
sudo apt-get install python-pip
sudo pip install flask
sudo pip install mysql-
   connector
```

Tanmay is an intern with the TLC, IIT Hyderabad. Durga is a UG student at IIT Hyderabad. *GVV Sharma is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in this manual is released under GNU GPL. Free and open source.

1.2 Testing Flask

Since installation of Flask is now complete, verify that flask is working by using the example below.

1) **Code:**

```
from flask import Flask
#Import the Flask class
app = Flask(__name__)
#Flask take (__name__)as an
   argument.
@app.route('/')
# '/' which url should call
   the associate function.
def student():
        return "Hello World"
if __name__=='__main__':
#server runs if the scripts
   executed directly from
   python interpreter and not
   used as an imported module.
        app.run()
# runs the application on
   local server
```

2) Save the file as **hello.py**.
3) open the terminal and run

```
python hello.py
```

An ip address will be displayed on the terminal.

4) Open the address on your favourite browser. ”Hello world” will be displayed

2 Mariadb

MariaDB Server is one of the most popular database servers in the world. The following installation instructions are for Ubuntu. Installation on other Linux systems are likely to be similar.

2.1 *Software Installation*

Refer to Link
https://www.liquidweb.com/kb/how-to-install-mariadb-5-5-on-ubuntu-14-04-lts/

1) Type the following commands on the terminal

```
sudo apt-get install software-
    properties-common
sudo apt-key adv --recv-keys --
    keyserver hkp://keyserver.
    ubuntu.com:80 0
    xcbcb082a1bb943db
sudo add-apt-repository 'deb␣
    http://mirror.jmu.edu/pub/
    mariadb/repo/5.5/ubuntu␣
    trusty␣main'
sudo apt-get update
sudo apt-get install mariadb-
    server
```

You may receive the following prompt or something similar:
After this operation, 116 MB of additional disk space will be used. Do you want to continue? [Y/n]
Enter Y to continue.
Next you will be asked:
New password for the MariaDB root user:
This is an administrative account in MariaDB with elevated privileges; enter a strong password.
Then you will be asked to verify the root MariaDB password:
Repeat password for the MariaDB root user:
That is it! Your basic MariaDB installation is now complete!
Be sure to stop MariaDB before proceeding to the next step:
sudo service mysql stop

2.2 *Configuration*

Configure and Secure MariaDB for Use

1) Now we will instruct MariaDB to create its database directory structure:
sudo mysql_install_db
2) Start MariaDB
sudo service mysql start
3) And now let us secure MariaDB by removing the test databases and anonymous user created by default:
sudo mysql_secure_installation
4) You will be prompted to enter your current password. Enter the root MariaDB password set during installation:
Enter current password for root (enter for none):
Then, assuming you set a strong root password, go ahead and enter n at the following prompt:
Change the root password? [Y/n] n
Remove anonymous users, Y:
Remove anonymous users? [Y/n] Y
Disallow root logins remotely, Y:
Disallow root login remotely? [Y/n] Y
Remove test database and access to it, Y:
Remove test database and access to it? [Y/n] Y
And reload privilege tables, Y:
Reload privilege tables now? [Y/n] Y
5) Verify MariaDB Installation
Check Version
mysql -V

3 Database Application

3.1 *Creating a Database*

1) Open the terminal and type

```
mysql -u root -p
```

You will be asked for a password. Enter it.
2) Create a database called test using the following command.

```
CREATE DATABASE Test;
```

3) In order to use the Database type

```
USE Test;
```

You will enter into the Database called Test.
4) Now create a table named test with parameters as Name and Roll Number.

```
CREATE TABLE test(name varchar
    (20) not null,roll varchar
    (20) not null);
```

varchar(20) means string of size 20 characters.

5) To see the format of the fields in **test**

```
desc test;
```

3.2 *Creating HTML Forms*

1) Type the following code in a file called **student.html** and open it using a browser. You will see boxes with **Name, Roll**. Also, there will be a button called **submit** and two links titled **Show List** and **Update**. the

```

<html>
<body>
  <form action ="/act" method="POST">
    <p>Name<input type ="text" name ="name"/></p>
        <p>Roll<input type =" text" name ="roll"/></p>
        <p><input type ="submit" value="submit"/></p>
        <p><a href="/display">Show List</a></p>
        <p><a href="/update">Update</a></p>
  </form>
</body>
</html>
```

2) Type the following code in a file called **message.html**. The purpose of this file is to display status messages.

```
<html>
<body>
        <p>output:{{msg}}</p>
</body>
</html>
```

3) Save both the html files in a folder called **templates**.

3.3 *Python Connector from Browser to Database*

1) Type the following code in a file called **store.py**.

```
from flask import Flask, render_template,request
import mysql.connector as mariadb
app=Flask(__name__)
@app.route('/')
def student():
  return render_template('student.html')

@app.route ('/act', methods =['GET','POST'])
def act():
  if (request.method == 'POST'):
        try:
          name=request.form['name']
          roll=request.form['roll']
          conn=mariadb.connect(user='root',password='123',database='Test')
          cur=conn.cursor()
          sql="INSERT INTO test(name,roll)values('{}','{}')".format(name,roll)
          cur.execute(sql)
          conn.commit()
          msg="Data Has Been Stored"
          return render_template('message.html',msg=msg)
        except:
          return "Database connection error"
if __name__=='__main__':
  app.run(debug = True)
```

2) Make sure that the python file is outside the **templates** directory. Now type

```
python store.py
```

on the terminal. An address will be displayed on the terminal.

3) Enter the above address in a browser. Fill the name and roll number and hit submit.

3.4 Fetching the stored Data from the Database

1) Save the following code in a file called **display.html**.

```
<html>
<body>
  <table border=1>
        <thead>
        <th>Name</th>
        <th>Roll</th>
        </thead>
        {% for row in rows %}
  <tr>
        <td>{{ row[0]}}</td>
        <td>{{ row[1]}}</td>
  </tr>
        {% endfor %}
  </table>
        <p><a href=”/”>Back To
           Home Page</a></p>
        <p><a href=”/update”>
           Update</a></p>
</body>
</html>
```

2) Save the following code in a file titled **display.py**.

3)

```
from flask import Flask,
   render_template,request
import mysql.connector as
   mariadb
app=Flask(__name__)
@app.route('/')
def list():
        conn=mariadb.connect(
           user='root',password
           ='123',database='
           Test')
        # Connecting to
           Database
        cur=conn.cursor()
        cur.execute(”Select␣*␣
           from␣test”) #This
           query is used to
           fetch the Data from
           the Database
        rows=cur.fetchall()
        return render_template(
           ”display.html”,rows=
           rows)
        # Returning display.
           html File
if __name__ == '__main__':
        app.run(debug = True)
```

4) Now open the terminal and type

```
python display.py
```

An address will be displayed.

5) Open this address in a browser. You can see all the Name and Roll No entries in the database.

3.5 Updating the Database

1)
2) Save the following code in a file with titled **show.html**.

```
<html>
<body>
  <table border=1>
        <td>Name</td>
        <td>Roll</td>
        <td>update</td>
        {% for row in rows %}
  <tr>
        <form action=”/
           testupdate” method=”
           POST”>
        <td><input type =”text”
            name =”name” value
           ={{ row[0]}}></td>
        <td><input type =”text”
            name =”roll” value
           ={{ row[1]}}></td>
        <td><input type = ”
           submit” value =”
           update”></td>
        </form>
        </tr>
        {% endfor %}
  </table>
</body>
</html>
```

3) Save the following code in a file titled **update.py**.

4)
```
from flask import Flask,
   render_template,request
import mysql.connector as
   mariadb
app=Flask(__name__)
@app.route('/')
def list():
        conn=mariadb.connect(
           user='root',password
           ='123',database='
           Test')
        # connecting to the
           database
        cur=conn.cursor()
        cur.execute("Select␣*␣
           from␣test")
        # fetching all the data
            from test table.
        rows=cur.fetchall()
        return render_template(
           "show.html",rows=
           rows)
        #returning show.html
           file

@app.route ('/testupdate',
   methods =['GET','POST'])
def testupdate():
        conn=mariadb.connect(
           user='root',password
           ='123',database='
           Test')
        cur=conn.cursor()
        name=request.form['name
           ']
        roll=request.form['roll
           ']
        print(roll)
        print(name)
        cur.execute("UPDATE␣
           test␣set␣roll='{}'␣
           where␣name='{}'".
           format(roll,name))
        # Query for updating
           the data in test
           table.
        conn.commit()
        return render_template(
           'message.html',msg="
           Data␣updated")
@app.route('/backhome')
def backhome():
        return render_template(
           'student.html')
        # returing to the main
           page after updating
if __name__ == '__main__':
        app.run(debug = True)
```

5) Now open the terminal and run the **update.py** file.
6) Update whatever data you wish to and click the Update button.
7) Run **display.py** to verify that your data is indeed updated.

3.6 Linking all modules to create the Database application

1) Save the following code in a file called **output.html**.

```
<html>
<body>
        <p>output:{{msg}}</p>
        <p><a href="/">Home</a>
           </p>
        <p><a href="/display">
           Show List</a></p>
        <p><a href="/update">
           Update</a></p>
</body>
</html>
```

2) Save the following code in a file titled **app.py**

```
from flask import Flask,
   render_template,request
import mysql.connector as
   mariadb
app=Flask(__name__)
@app.route('/')
def student():
  return render_template('
     student.html')
@app.route ('/act', methods = [
   'GET','POST'])
def act():
  if (request.method == 'POST')
     :
        try:
            name=request.form['
                name']
            roll=request.form['
                roll']
            conn=mariadb.connect(
                user='root',
                password='123',
                database='Test')
            cur=conn.cursor()
            sql="INSERT INTO test
                (name,roll) values
                ('{}','{}')".
                format(name,roll)
            cur.execute(sql)
            conn.commit()
            return
                render_template("
                output.html",msg="
                Data Has Been
                Stored")
        except:
            return "Database
                connection error"
@app.route('/display')
def display():
   conn=mariadb.connect(user='
       root',password='123',
       database='Test')
   cur=conn.cursor()
   cur.execute("Select * from
       test")
   rows=cur.fetchall()
   return render_template("
       display.html",rows=rows)
@app.route('/update')
def list():
   conn=mariadb.connect(user='
       root',password='123',
       database='Test')
   cur=conn.cursor()
   cur.execute("Select * from
       test")
   rows=cur.fetchall()
   return render_template("show.
       html",rows=rows)

@app.route ('/testupdate',
    methods =['GET','POST'])
def testupdate():
   conn=mariadb.connect(user='
       root',password='123',
       database='Test')
   cur=conn.cursor()
   name=request.form['name']
   roll=request.form['roll']
   print(roll)
   print(name)
   cur.execute("UPDATE test set
       roll='{}' where name='{}'"
       .format(roll,name))
   conn.commit()
   return render_template('
       student.html',msg="Data
       updated")
@app.route('/backhome')
def backhome():
   return render_template('
       student.html')

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

3) Run **app.py**
4) Start using your application.
5) Modify your application so that you may delete a record.