

Topic 7

Front-end Application Development

Dr Diarmuid Ó Briain



Licence



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
Full License: <http://creativecommons.org/licenses/by-sa/4.0>

Learning objectives

By the end of this topic the learner will be able to:

- Develop a front-end that interface to manage a back-end database.

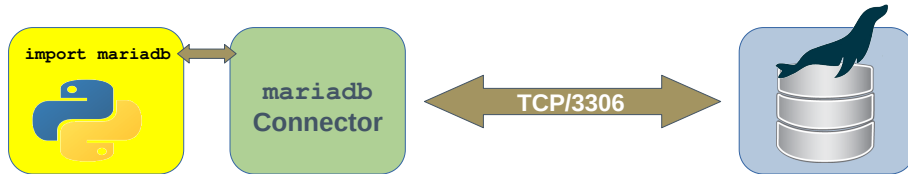
MariaDB Connector/Python



MariaDB

MariaDB Connector/Python

- The MariaDB connector/Python permits the use of Python to manage data stored in MariaDB Platform.



```
conn = mariadb.connect(
    user=, password=,
    host=, port=,
    database=)
cur = conn.cursor()
cur.execute(query)
```

- connect ()** : Control access to the database.
- cursor ()** : Keep track of where an operation is taking place in the database.

MariaDB Connector/Python

```
~$ ./basic_query.py
#!/usr/bin/env python3
```

```
import mariadb
```

```
query = ("SELECT * FROM EngProject;")
conn = mariadb.connect(user="enguser", password="engpass",
                      host="127.0.0.1", port=3306, database="Eng")
```

```
cur = conn.cursor()
cur.execute(query)
print(cur.fetchall())
cur.close()
conn.close()
```

```
~$ ./basic_query.py
```

```
[(1, 'alovelace', 'Ada', 'Lovelace', 'ada@lovelace.com', 'Programmer'), (2,
'lmenabrea', 'Luigi', 'Menabrea', 'luigi@menabrea.it', 'Politician'), (3,
'equinn', 'Edel', 'Quinn', 'edel@quinn.net', 'Nurse'), (4, 'vcunnane',
'Vincent', 'Cunnane', 'vc@tus.ie', 'Professor')]
```

Default Data Generator

- Python program to clear and populate database with data.

```
~$ tree
```

```
.
├── conf.yml
└── default_dbsetup.py
```

```
~$ cat conf.yml
```

```
---
# Configuration

user: enguser
host: 127.0.0.1
port: 3306
database: Eng
table: EngProject
...
```

Default Data Generator - Extracts

- Read configuration from `conf.yml`

```
# // Get the configuration information from YAML //
with open(f"{__dir__}/conf.yml", "r") as fh:
    try:
        conf = yaml.safe_load(fh)
        table = conf.pop("table")
        time = conf.pop("time")
    except yaml.YAMLError as err:
        print(f"Error: {err}", file=sys.stderr)
        sys.exit(1)

# // Test insert lines //
print(f"\ntable: {table}\nconf: {conf}")
exit()
```

Default Data Generator - Extracts

- Test the read configuration from `conf.yml`

```
data_gen~$ ./default_dbsetup.py
table: EngProject
conf: {'user': 'enguser', 'host': '127.0.0.1', 'port': 3306, 'database': 'Eng'}
```

Default Data Generator - Extracts

- Use `mariadb` module to connect to the database

```
# // Connect to the MariaDB database //
try:
    conn = mariadb.connect (
        user=conf["user"],
        password=conf["password"],
        host=conf["host"],
        port=conf["port"],
        database=conf["database"])
    cur = conn.cursor()
    print(f"\nConnected to the {conf['database']} database\n")

except mariadb.Error as e:
    print(f"\nError connecting to MariaDB Platform: {e}")
    sys.exit(1)
```

Default Data Generator - Extracts

- Delete existing and add new tables

```
# // Delete the current database tables in MariaDB //
for key in db_tables.keys():
    query = f"DROP TABLE IF EXISTS Eng.{key};"
    cur.execute(query)
    print(query)

# // Create tables //
for query in db_tables.values():
    cur.execute(query)
    print(query)

# // Populate tables with some data //
for key, value in db_data.items():
    for str_ in value:
        query = f"INSERT INTO {key} VALUES ({str_})"
        cur.execute(query)
        conn.commit()
        print(query)
```

Default Data Generator

```
data_gen~$ ./default_dbsetup.py
Default engCORE Database generator
-----

This program drops existing 'EngProject' and 'EngHobbies'
tables from the 'Eng' database, creates net tables and
populates them with some sample data.

Enter the password to access database: engpass

Connected to the Eng database

DROP TABLE IF EXISTS Eng.EngProject;
DROP TABLE IF EXISTS Eng.EngHobbies;
CREATE TABLE EngProject (Student_no INT NOT NULL, Username TEXT NULL, FirstName TEXT NULL, LastName TEXT NULL, Email
TEXT NULL, Role TEXT NULL, PRIMARY KEY (student_no));
CREATE TABLE EngHobbies (Student_no INT NOT NULL, Hobbies TEXT NULL, PRIMARY KEY (Student_no));
INSERT INTO EngProject VALUES (000000, 'cbabage', 'Charles', 'Babbage', 'charles@babbage.com', 'Hardware')
INSERT INTO EngProject VALUES (000001, 'alovelace', 'Ada', 'Lovelace', 'ada@lovelace.com', 'Programmer')
INSERT INTO EngProject VALUES (000002, 'lmenabrea', 'Luigi', 'Menabrea', 'luigi@menabrea.it', 'Politician')
INSERT INTO EngHobbies VALUES (000000, 'cricket, cards')
INSERT INTO EngHobbies VALUES (000001, 'camogie, horses')
INSERT INTO EngHobbies VALUES (000002, 'soccer, pasta')

Database Eng has now been populated.
```



mariadb_conn.py



mariadb_conn.py

```
~$ python3
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> sys.path
['', '/usr/lib/python310.zip', '/usr/lib/python3.10',
'/usr/lib/python3.10/lib-dynload', '/home/ada/.local/lib/python3.10/site-
packages', '/usr/local/lib/python3.10/dist-packages',
'/usr/lib/python3/dist-packages']
>>> quit()
```

```
~$ cp mariadb_conn.py /usr/lib/python3/dist-packages
[sudo] password for ada: ada_pass
```

```
~$ sudo chown root: /usr/lib/python3/dist-packages/mariadb_conn.py
~$ sudo chmod 644 /usr/lib/python3/dist-packages/mariadb_conn.py
```



mariadb_conn.py

```
~$ python3
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import mariadb_conn
>>> help(mariadb_conn)

help on module mariadb_conn:

NAME
    mariadb_conn - MariaDB connector program for SQL MasterClass

FUNCTIONS
    db_connect(**kwargs)
        MariaDB Connection

    db_delete(conn, cur, query)
        MariaDB DELETE Query

    db_insert(conn, cur, query)
        MariaDB INSERT Query

    db_select(cur, table)
        MariaDB SELECT Query

FILE
    /home/ada/.local/lib/python3.9/site-packages/mariadb_conn.py
```



mariadb_conn.py

```
~$ ./mariadb_conn.py
Connected to the EngDatabase

1. SELECT      2. INSERT      3. DELETE      q. QUIT      Test What? : 1

[('Student_no', 'Username', 'FirstName', 'LastName', 'Email', 'Role'),
(0, 'alovelace', 'Ada', 'Lovelace', 'ada@lovelace.com', 'Programmer'),
(1, 'cbabage', 'Charles', 'Babbage', 'charles@babbage.com', 'Hardware')]

1. SELECT      2. INSERT      3. DELETE      q. QUIT      Test What? : 2
Successfully inserted into the EngProject Database

[('Student_no', 'Username', 'FirstName', 'LastName', 'Email', 'Role'),
(0, 'alovelace', 'Ada', 'Lovelace', 'ada@lovelace.com', 'Programmer'),
(1, 'cbabage', 'Charles', 'Babbage', 'charles@babbage.com', 'Hardware'),
(3, 'diddly', 'Diddly', 'Dee', 'dee@diddly.com', 'Programmer')]

1. SELECT      2. INSERT      3. DELETE      q. QUIT      Test What? : 3
Successfully deleted from the EngProject Database

[('Student_no', 'Username', 'FirstName', 'LastName', 'Email', 'Role'),
(0, 'alovelace', 'Ada', 'Lovelace', 'ada@lovelace.com', 'Programmer'),
(1, 'cbabage', 'Charles', 'Babbage', 'charles@babbage.com', 'Hardware')]

1. SELECT      2. INSERT      3. DELETE      q. QUIT      Test What? : q

Testing completed!!
```



Python Virtual Environment



The database connector

Install the Virtual Environment feature.

```
~$ sudo apt install -y python3-venv
```

Create a Virtual environment in the directory ~/.venv.

```
~$ python3 -m venv ~/.venv
```

Activate the virtual Environment.

```
~$ source ~/.venv/bin/activate  
(.venv)~$ python3  
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.
```

Find the path to the modules in the Virtual Environment and note the virtual environment path to packages.

```
>>> import sys  
>>> sys.path  
['', '/usr/lib/python39.zip', '/usr/lib/python3.9', '/usr/lib/python3.9/lib-  
dynload', '/home/ada/.venv/lib/python3.10/site-packages']  
>>>
```



The database connector

The following three Python modules are installed by default when the virtual environment is created.

```
(.venv)~$ python3 -m pip list  
Package      Version  
-----  
pip          20.3.4  
pkg-resources 0.0.0  
setuptools   44.1.1
```

Install the following Python modules.

```
(.venv)~$ python -m pip install flask jinja2 mariadb pyyaml
```

Move mariadb_conn.py to the venv path.

```
(.venv)~$ cp ~/web_main/mariadb_conn.py .venv/lib/python3.9/site-packages/
```

After install.

```
(.venv)~$ python3 -m pip list  
Package      Version  
-----  
blinker      1.7.0  
click        8.1.7  
Flask        3.0.0  
itsdangerous 2.1.2  
Jinja2       3.1.2  
mariadb      1.1.8  
MarkupSafe   2.1.3  
packaging    23.2  
pip          22.0.2  
PyYAML       6.0.1  
setuptools   59.6.0  
Werkzeug     3.0.1
```



Web Application Development



Custom database interface

```
(.venv)~$ tree ~/web_main
/home/ada/web_main
├── app.wsgi
├── conf.yml
├── init.py
├── README.txt
├── static
│   ├── css
│   │   └── main.css
│   ├── images
│   │   └── TUS_White.png
├── templates
│   ├── about.html
│   ├── delete.html
│   ├── home.html
│   ├── index.html
│   ├── layout.html
│   ├── login.html
│   ├── read.html
│   └── write.html
└── tools
    ├── default_dbsetup.py
    └── mariadb_conn.py
```

```
(.venv)~$ python3 ~/web_main/init.py
* Serving Flask app 'init'
* Debug mode: on
WARNING: This is a development server. Do not use it
in a production deployment. Use a production WSGI
server instead.
```

```
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 460-699-611
```

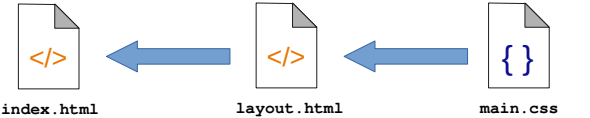
```
# // Flask app //
app = Flask(__name__)

@app.route("/")
def index():
    """The main index page"""
    return render_template("index.html", title=title)
```

The creation of index.html

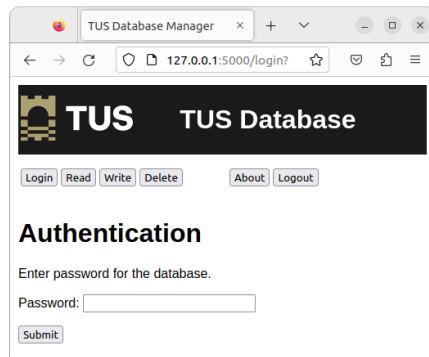
Extract from: init.py
title = "TUS Database"

```
@app.route("/")
def index():
    """The main index page"""
    return render_template("index.html", title=title)
```



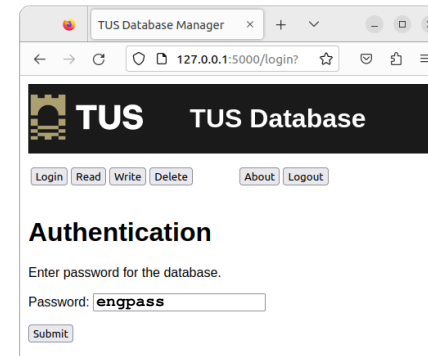
```
~$ cat index.html
{% include "layout.html" %}
{% block content %}
<div>
<h1>{{ title }}</h1>
<p>This is a utility to manage the {{ title }}.</p>
<p>Login with the password to access the database.</p>
</div>
{% endblock %}
```

Custom database interface



```
127.0.0.1 - - [15/Oct/2022 14:31:18] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Oct/2022 14:31:18] "GET /static/css/main.css HTTP/1.1" 200 -
127.0.0.1 - - [15/Oct/2022 14:31:18] "GET /static/images/TUS_White.png HTTP/1.1" 200 -
127.0.0.1 - - [15/Oct/2022 14:31:18] "GET /favicon.ico HTTP/1.1" 404 -
```

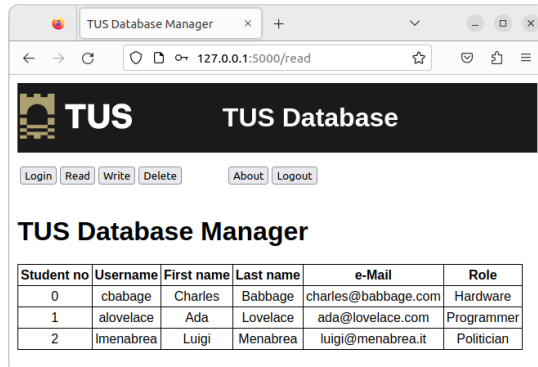
Login



```
@app.route("/login")
def login():
    """Page to attract login password from the user"""
    msg = "Enter password for the database."
    return render_template("login.html", title=title, msg=msg)
```

Read from the database

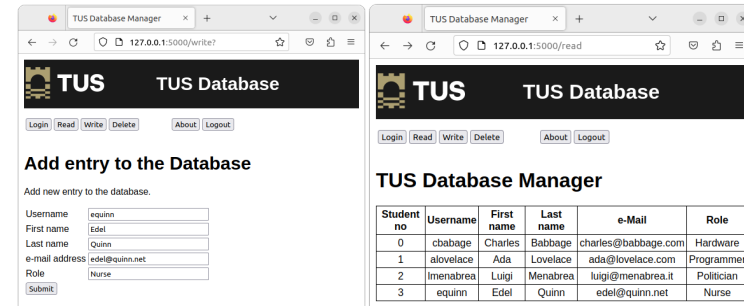
```
SHOW * FROM 'EngProject';
```



Student no	Username	First name	Last name	e-Mail	Role
0	cbabage	Charles	Babbage	charles@babbage.com	Hardware
1	alovelace	Ada	Lovelace	ada@lovelace.com	Programmer
2	lmenabrea	Luigi	Menabrea	luigi@menabrea.it	Politician

Write to the database

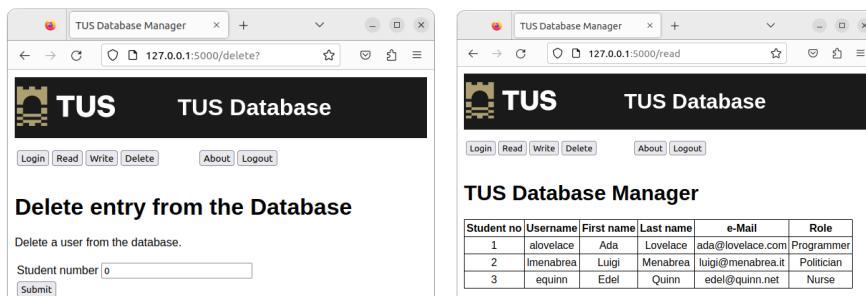
```
INSERT INTO 'EngProject' VALUES(3, 'equinn', 'Edel', 'Quinn', 'eden@quinn.net', 'Nurse');
```



Student no	Username	First name	Last name	e-Mail	Role
0	cbabage	Charles	Babbage	charles@babbage.com	Hardware
1	alovelace	Ada	Lovelace	ada@lovelace.com	Programmer
2	lmenabrea	Luigi	Menabrea	luigi@menabrea.it	Politician
3	equinn	Edel	Quinn	eden@quinn.net	Nurse

Delete from the database

```
DELETE FROM EngProject WHERE 'Student_no'=0;
```



Student no	Username	First name	Last name	e-Mail	Role
1	alovelace	Ada	Lovelace	ada@lovelace.com	Programmer
2	lmenabrea	Luigi	Menabrea	luigi@menabrea.it	Politician
3	equinn	Edel	Quinn	eden@quinn.net	Nurse



Web Application
In Production

TUS

Python 3

Flask

Apache

Moving the database into production on Apache2

- Move application into the Apache2 server root and give the webserver ownership of the files

```
~$ sudo mv ~/web /var/www/html/
~$ sudo chown -R www-data: /var/www/html/web

~$ sudo usermod -a -G ada www-data
~$ sudo usermod -a -G www-data ada

~$ cat /etc/group | grep ^www
www-data:x:33:ada
~$ cat /etc/group | grep ^ada
ada:x:1000:www-data
```

Moving the database into production on Apache2

- Create an Apache2 site configuration file

```
~$ cd /etc/apache2/sites-available/

/etc/apache2/sites-available$ sudo mv 000-default.conf 000-default.conf.orig
/etc/apache2/sites-available$ cat <<EOM | sudo tee 000-default.conf

<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    WSGIScriptAlias / /var/www/html/web/app.wsgi
    <Directory /var/www/html/web>
        Order allow,deny
        Allow from all
    </Directory>

    ErrorLog /error.log
    CustomLog /access.log combined

</VirtualHost>
EOM
```

Moving the database into production on Apache2

- The Apache2 site configuration file points to the WSGI Script Alias `/var/www/html/web/app.wsgi`
- This file inserts the `venv` module path as well as the root to the web directory into the Apache2, python path
- It then calls `app`, the instance of `Flask()` from `init.py`

```
~$ cat app.wsgi
import sys

sys.path.insert(0, "/home/ada/.venv/lib/python3.10/site-packages")
sys.path.insert(0, "/var/www/html/web")

from init import app as application
```

Install WSGI and enable it for Apache2

```
~$ sudo apt install libapache2-mod-wsgi-py3
~$ sudo a2enmod wsgi
Enabling module wsgi.
```

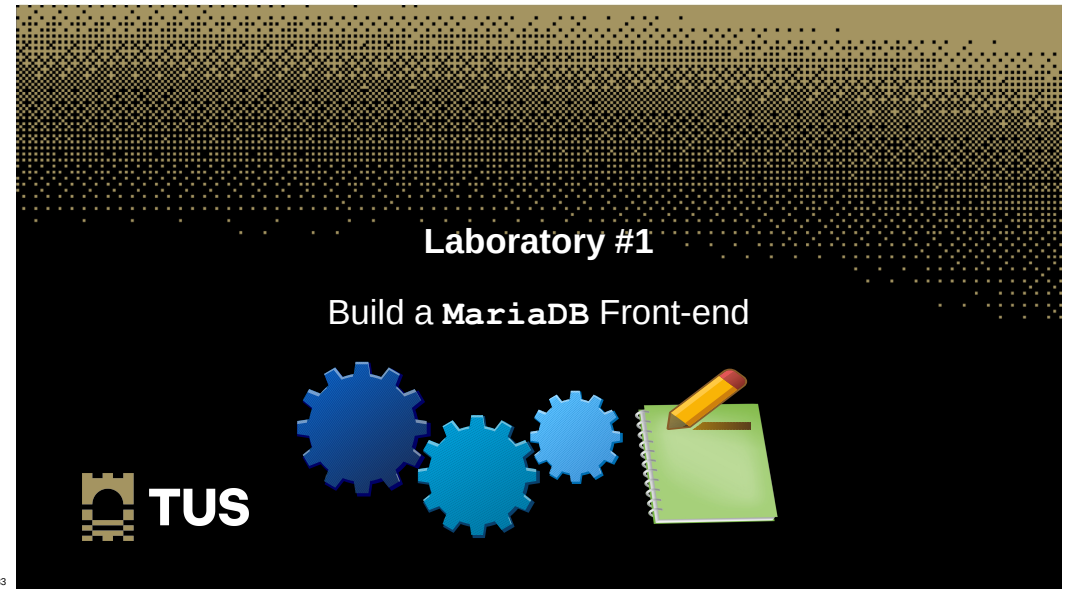
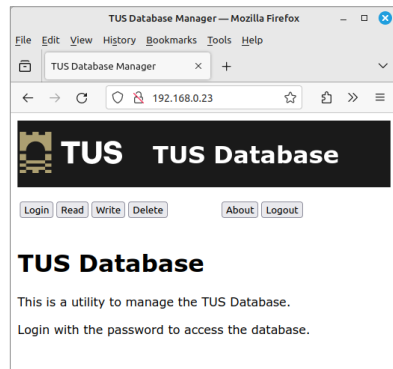
- Relaunch Apache2 to enable

```
~$ sudo systemctl restart apache2

~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-09-14 23:09:13 IST; 29s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 3751 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 3755 (apache2)
    Tasks: 55 (limit: 9493)
   Memory: 23.5M
      CPU: 66ms
```


Test the Apache2 Service

- From another workstation test the service is operational.



Laboratory #1 – Create database front-end

Create a custom interface to the counties database

- Create a web based interface to the database.
- Permit input of County capitals, rivers, mountains and sports.
- Do not permit removal of county information from the Counties table once entered.
- Have a mechanism to change the county sport if it is wrong.
- Document each stage.

Learning Objectives

- Develop a front-end that interface to manage a back-end database. ✓



TUS

Oibiceil Teicneolaíochta na Sionainne:
Lár Tíre, An Bhaile Thiar Láir
Technological University of the Shannon
Midlands Midwest

EUR ING Dr Diarmuid Ó Briain
Innealtóir Cairte agus
Léachtóir Sinsearach

✉ diarmuid.obriain@tus.ie | www.tus.ie
Campas Maolais, Páirc Maolais,
Luimneach, V94 EC5T, Éire



Thank you



TUS

