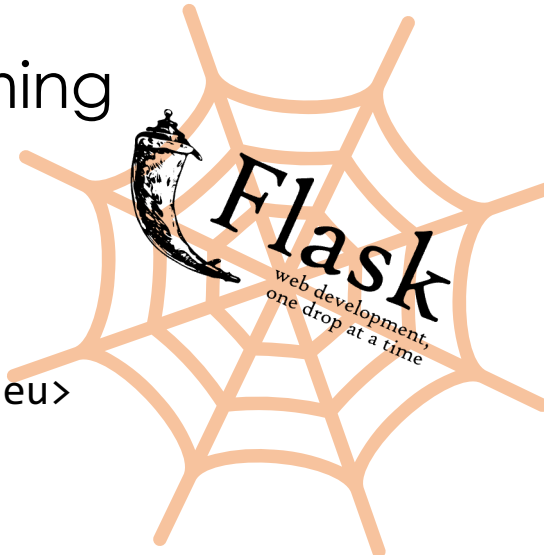
TELECOM nancy
Ingénieurs du numérique • Inspiring your digital future

UNIVERSITÉ DE LORRAINE

Web / BD

Introduction to Web Programming

2025-2026



Gérald Oster <gerald.oster@telecomnancy.eu>

Data Persistance

- Put everything in a database!

SQLite3

- SQLite3 (https://www.sqlite.org/):
  - « Public domain » Embedded database / SQL92 compliant (mostly)
- In a Linux (Debian/Ubuntu) machine :

```
$ sudo apt-get install sqlite3
```

- Main commands:

  `.help` (your friends ;b)

  `.open` (open a database file – sqlite file format)

  `.quit` (exit SQLite)

  `.tables` (show all tables)

  `.schema` (show schema of all tables)

SQLite3

```
$ sqlite3 halloween.db
SQLite version 3.39.5 2022-10-14 20:58:05
Enter ".help" for usage hints.

sqlite> CREATE TABLE monsters(id integer, name varchar, height integer);
sqlite> .schema
CREATE TABLE monsters(id integer, name varchar, height integer);

sqlite> INSERT INTO monsters VALUES (1, 'Cthulhu', 47);
sqlite> INSERT INTO monsters VALUES (2, 'Nyarlathotep', 100);
sqlite> SELECT * FROM monsters ORDER BY height ASC;
1|Cthulhu|47
2|Nyarlathotep|100

sqlite> SELECT * FROM monsters ORDER BY height DESC;
2|Nyarlathotep|100
1|Cthulhu|47
```

Python / SQLite3

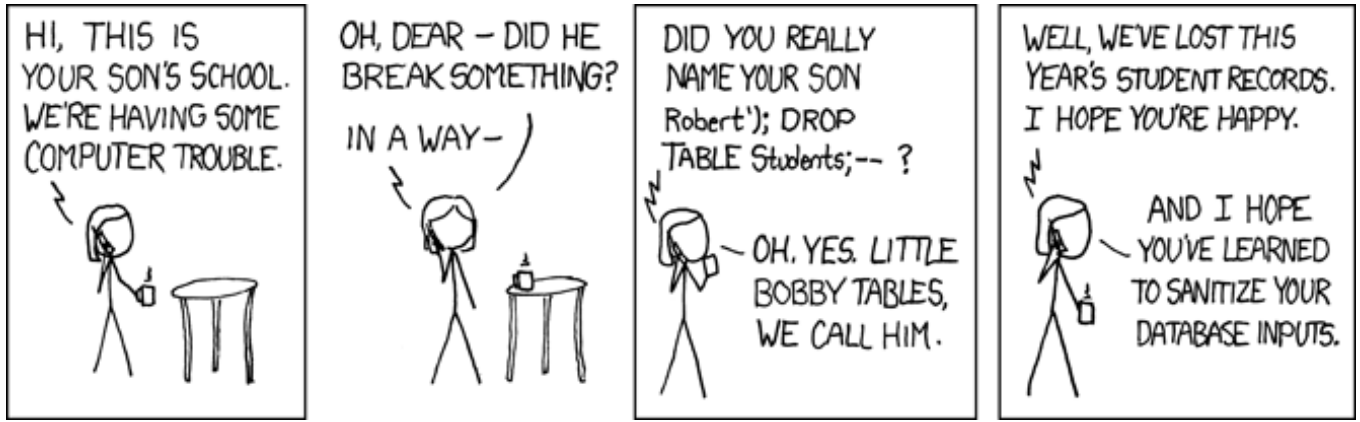
- https://docs.python.org/3/library/sqlite3.html

```
import sqlite3

con = sqlite3.connect('halloween.db')

cur = con.cursor()
# print(cur.execute('SELECT * FROM monsters')
# print(cur.fetchone())
# print(cur.fetchall())
for row in cur.execute('SELECT * FROM monsters ORDER BY height DESC’):
#   print(row)
    print(f'id: {row[0]}, name: {row[1]}, height: {row[2]}')

con.commit()
con.close()
```

Python / SQLite3 (cont.)

- What not to do (security issues: SQL injection):

```
monster_name = 'Cthulhu'
cur.execute("SELECT * FROM monsters WHERE name = '%s'" % monster_name)
# or:
cur.execute(f"SELECT * FROM monsters WHERE name = '{monster_name}'")
```



[Source: https://xkcd.com/327/]

Python / SQLite3 (cont.)

- How to do it:

```
# either:
cur.execute("SELECT * FROM monsters WHERE (?)", ('Cthulhu',))
# or:
cur.execute("SELECT * FROM monsters WHERE name=:mname", {"mname": 'Cthulhu'})
```

Beware! `('Cthulhu')` is `'Cthulhu'` so it is a string!
but you need to pass a python tuple!
`('Cthulhu',)` is a tuple in Python

A Simple App using Flask / SQLite3

```
from flask import Flask
from flask import g
import sqlite3

app = Flask(__name__)

DATABASE = 'halloween.db'

def get_db():
     db = getattr(g, '_database', None)
     if db is None:
          db = g._database = sqlite3.connect(DATABASE)
     return db

@app.teardown_appcontext
def close_connection(exception):
     db = getattr(g, '_database', None)
     if db is not None:
          db.close()
```

A Simple App using Flask / SQLite3 (cont.)

…

```
@app.route('/')
def status():
     return 'The server is running!'

@app.route('/monsters')
def all_monsters():
     c = get_db().cursor()
     c.execute("select * from monsters")
     return c.fetchall()

@app.route('/monsters/<int:monster_id>')
def one_monster(monster_id: str):
     c = get_db().cursor()
     c.execute("select * from monsters where id = ?", (monster_id,))
     return list(c.fetchone())
```

SQLAlchemy

https://www.sqlalchemy.org/

```
$ pip install SQLAlchemy
```

```
from sqlalchemy import create_engine, text

engine = create_engine('sqlite:///halloween.db')
# engine = create_engine('postgresql://user:password@host/database')

con = engine.connect()

rs = con.execute(text('SELECT * FROM book'))

for row in rs:
    print(row)

con.close()
```

SQLAlchemy / PostgreSQL

OPTIONAL

```
from sqlalchemy import create_engine, text

engine = create_engine('postgresql://postgres:mysecret@localhost/postgres')
con = engine.connect()

rs = con.execute(text('SELECT * FROM monsters'))
for row in rs:
   print(row)

con.close()
```

SQLAlchemy (cont.)

```
from sqlalchemy import create_engine, text

engine = create_engine('sqlite:///bookstore.db')
con = engine.connect()

rs = con.execute(text('DROP TABLE IF EXISTS book'))
rs = con.execute(text('CREATE TABLE book (id INTEGER PRIMARY_KEY,
                        title VARCHAR, primary_author VARCHAR)'))

statement = text('INSERT INTO book(id, title, primary_author) VALUES
                        (:id, :title, :primary_author)')
rs = con.execute(statement, {'id':1, 'title':'The Silmarillion',
                                    'primary_author':'Tolkien' })
for row in con.execute(text('SELECT * FROM book')):
   print(row)
con.close()
```

Flask-SQLAlchemy

https://flask-sqlalchemy.palletsprojects.com/en/3.1.x/

```
$ pip install flask_sqlalchemy
```

```
from flask import Flask
from sqlalchemy import text
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:////tmp/halloween.db'
db = SQLAlchemy(app)

@app.route('/')
def status():
    return 'The server is running!'

@app.route('/monsters')
def all_monsters():
    c = db.session
    res = c.execute(text("select * from monsters"))
    return str(res.fetchall())
```

PostgreSQL (using Docker ;§)

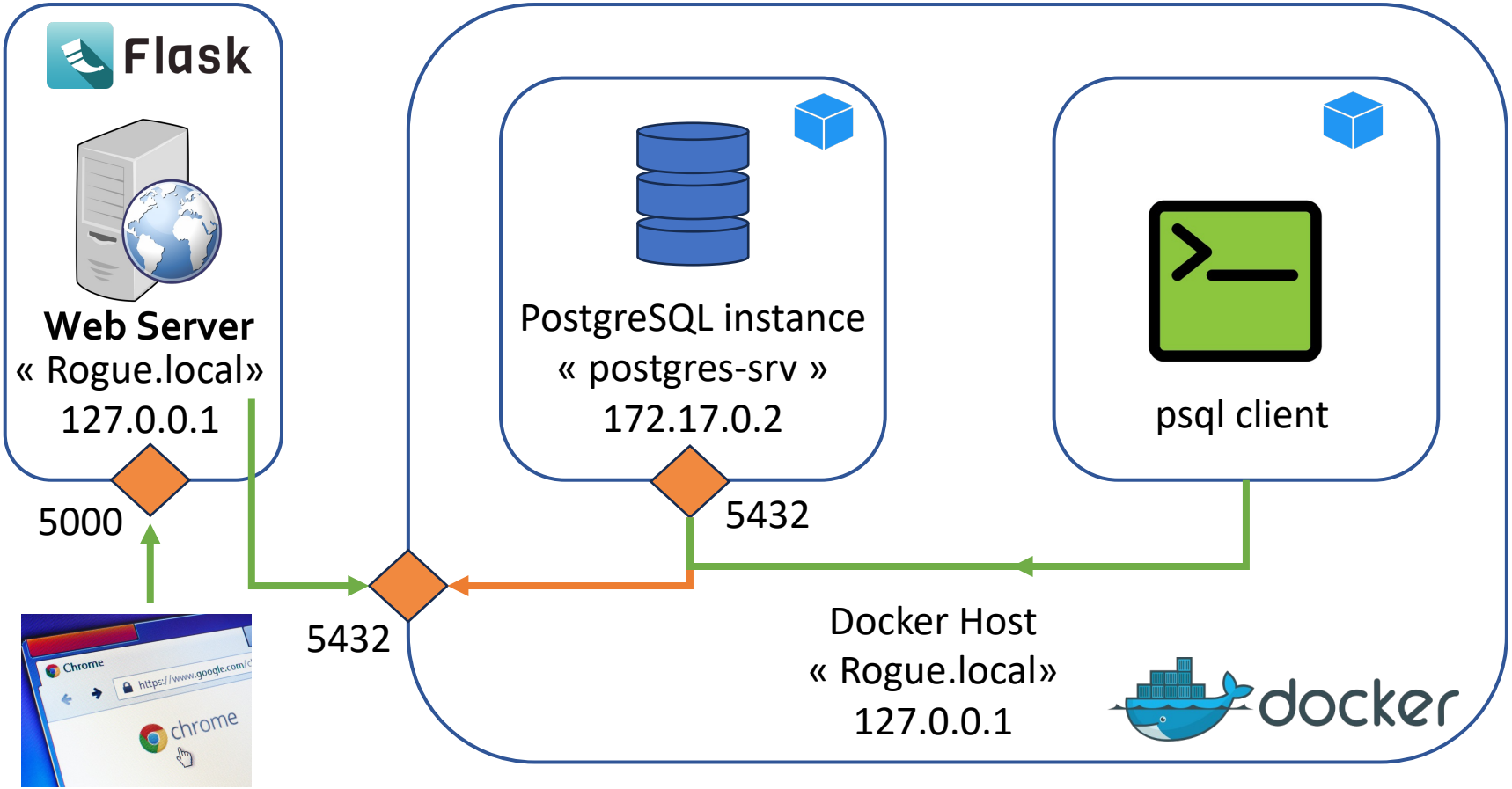
.SERVER in a container.

```
$ docker pull postgres:16
$ docker network create mynetwork
$ docker run –p 5432:5432 --name postgres-vm --network mynetwork
     --network-alias postgres-srv -e POSTGRES_PASSWORD=mysecret -d postgres
```

.CLIENT in a container.

```
$ docker ps
$ docker inspect postgres-vm | grep 'IPAddress'
 "IPAddress": "172.17.0.2",
$ docker run -it --network mynetwork --rm postgres psql -h postgres-srv -U
postgres
Password for user postgres:
psql (13.2 (Debian 13.2-1.pgdg100+1))
Type "help" for help.
postgres=# CREATE TABLE monsters(id integer, name varchar, height integer);
CREATE TABLE
postgres=# INSERT INTO monsters VALUES (1, 'Cthulhu', 47);
INSERT 0 1
```

Big Picture 😲


Flask
Web Server
« Rogue.local»
127.0.0.1
5000
PostgreSQL instance
« postgres-srv »
172.17.0.2
5432
psql client
5432
Docker Host
« Rogue.local»
127.0.0.1
docker
Chrome
https://www.google.com/
chrome

To go further…

- Me want Cookie!