# LAB 4-5

## I. Thông tin chung

- Thực hành chương 3: Crawling website, .
- Sinh viên lưu file với tên theo định dạng sau: Lab4_MSSV.ipynb,
- Sinh viên nộp lên Elearning của lớp học tại buổi thực hành 4.
- Deadline: 17h00, ngày 01/03/2023

## II. Yêu cầu

- Sử dụng các IDE (anaconda, pycharm, google colab,…) để tạo file thực hành.

## III. Nội dung

### 1. Chuẩn bị

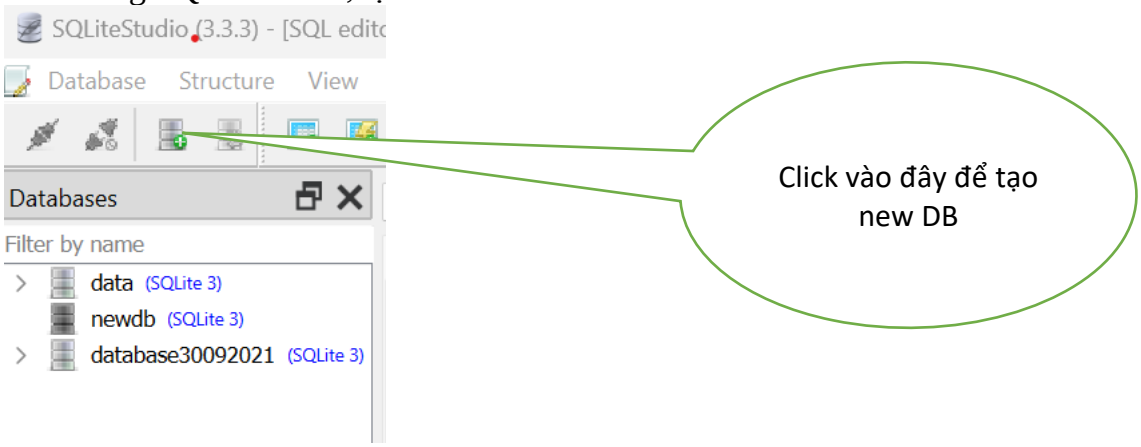
### 1.1. Cài đặt thư viện

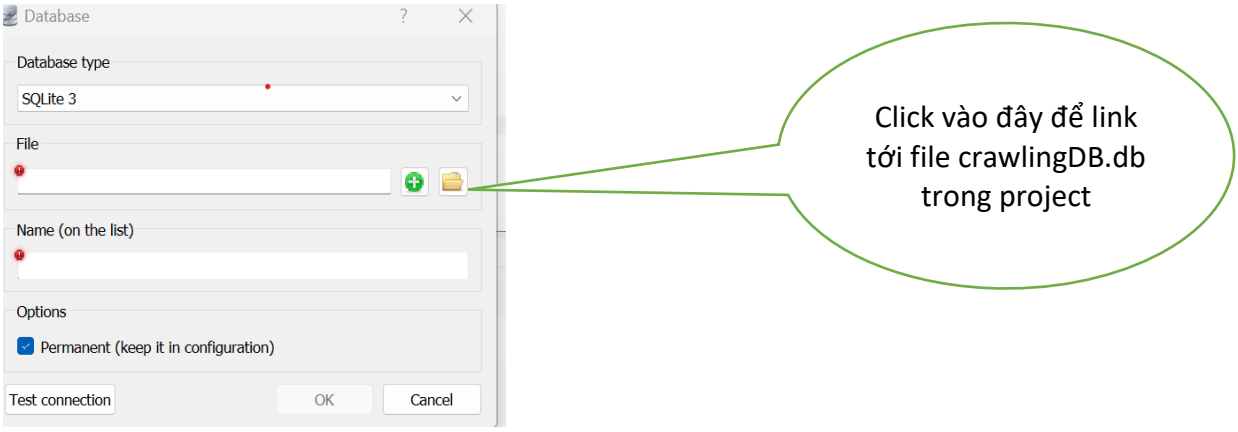
- Cài đặt sqlite3 studio
- Tạo project mới
- Install thư viện: thực hiện install hai thư viện sau
  - pip install flask
  - pip install newspaper3k #dùng để crawl data từ danh mục web

### 1.2. Thực hiện

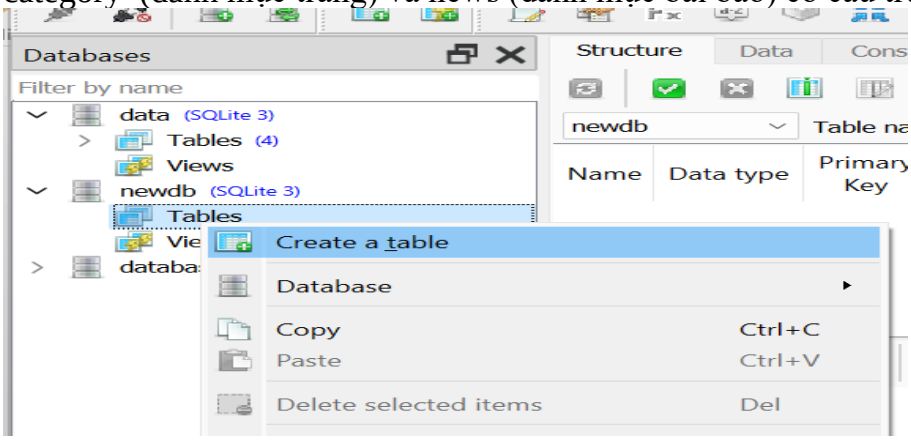
B1. Tạo thư mục DATA trong project => tạo file database có tên crawlingDB.db

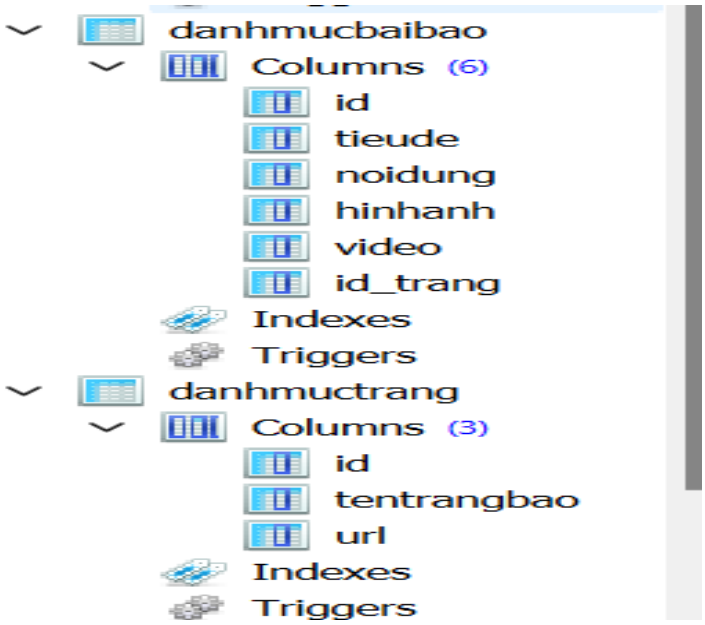
B2. Trong SQLiteStudio, tạo new DB



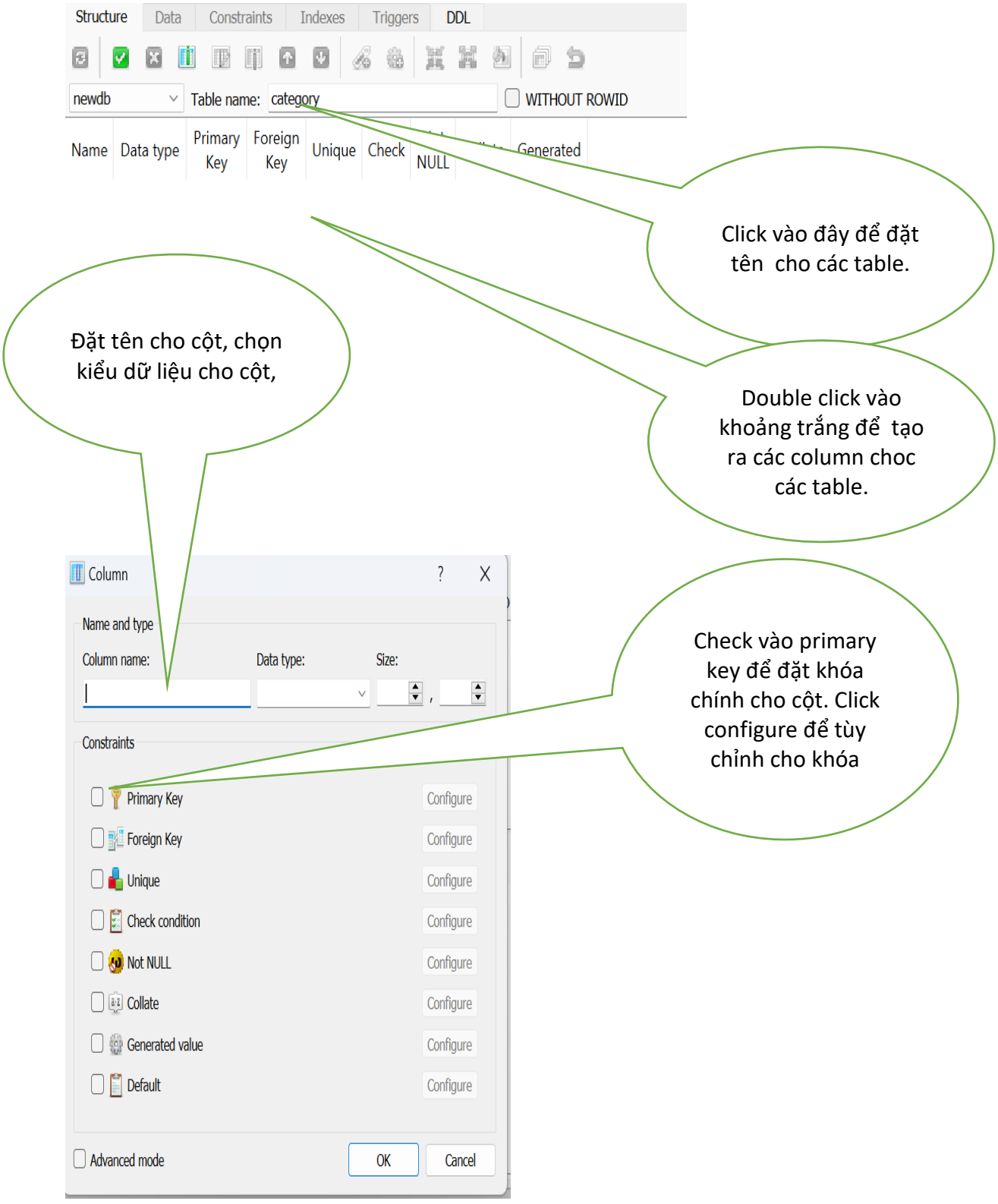


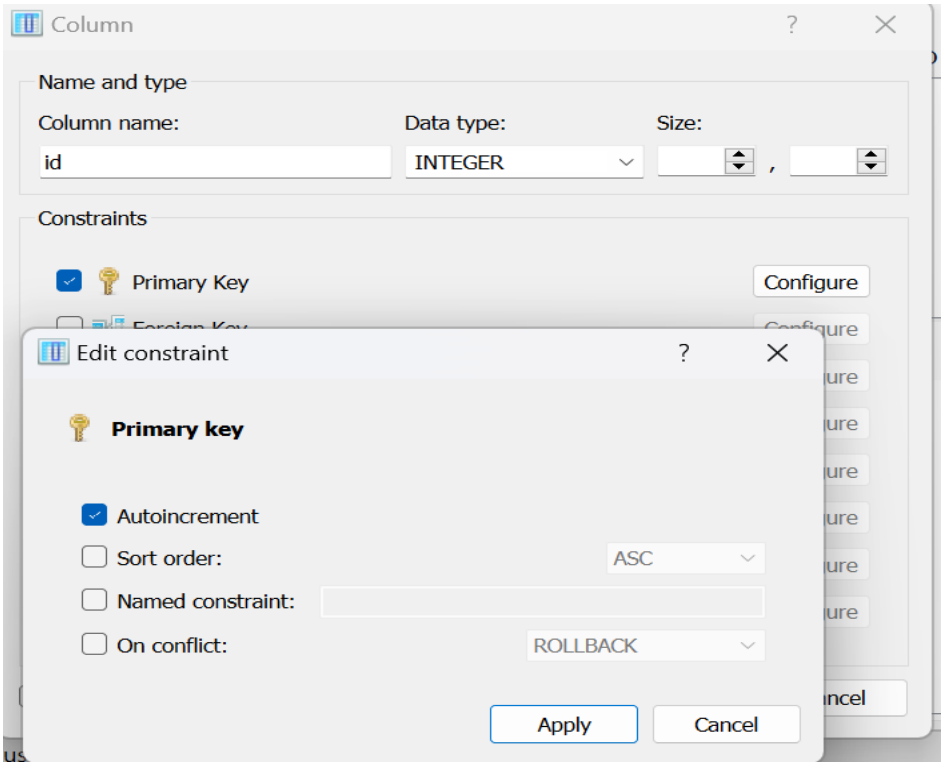

Trong new DB, click chuột phải vào tables => create table để tạo ra hai bảng category (danh mục trang) và news (danh mục bài báo) có cấu trúc như sau:




Để tạo các column cho table, thực hiện:


Structure
Data
Constraints
Indexes
Triggers
DDL
newdb
Table name:
category
WITHOUT ROWID
Name
Data type
Primary Key
Foreign Key
Unique
Check
NULL
Generated
Click vào đây để đặt tên cho các table.
Đặt tên cho cột, chọn kiểu dữ liệu cho cột,
Double click vào khoảng trắng để tạo ra các column choc các table.
Column
Name and type
Column name:
Data type:
Size:
Constraints
Primary Key
Foreign Key
Unique
Check condition
Not NULL
Collate
Generated value
Default
Configure
Advanced mode
OK
Cancel
Check vào primary key để đặt khóa chính cho cột. Click configure để tùy chỉnh cho khóa




mở bảng category =>thêm danh mục các trang báo sẽ tiến hành craw dl.

B3.Tạo hai module:

- api.py: để define các API, import các thư viện sau

```
from flask import Flask, jsonify, render_template, request
import utils
import render_templates
```

- utils.py: import các thư viện

```
from flask import Flask, jsonify
import newspaper
from newspaper import Article
import sqlite3
```

- Define các hàm trong utils

**get_all():**

```
def get_all(query):
    conn = sqlite3.connect("DATA/crawlingDB.db")
    data = conn.execute(query).fetchall()
    conn.close()

    return data
```

**add_news()**

```
#define function Add bài báo được craw từ web vào database, sử dụng newspaper3k
#tạo query để insert bài báo craw được vào table news,
#gọi conn.commit() để add vào database
def add_news(conn,url,category_id):
    query = """
    INSERT INTO news(subject,description,image,original_url,category_id)
    VALUES(?,?,?,?,?)
    """
    article = Article(url)
    article.download()
    article.parse()

    conn.execute(query, (article.title,article.text,article.top_image,article.url,category_id))
    conn.commit()
```

**get_news_url()**

```
# API lấy danh mục bài báo sử dụng newspaper3k
#cats: gọi function get_all để lấy tất cả danh mục có trong category
# tạo connection tới database
# với mỗi category được lấy, sẽ tiến hành lấy id, url, sau đó gọi method build của
newspaper để build link.
# với mỗi link sẽ gọi function add_news để add nội dung bài báo craw được vào database,
# sử dụng try - except để bỏ qua một số trường hợp không thể parse.
def get_news_url():
    cats = get_all("SELECT * FROM category")
    conn = sqlite3.connect("DATA/crawlingDB.db")
```

```
        for cat in cats:
            cat_id = cat[0]
            url = cat[2]
            cat_paper = newspaper.build(url)
            for article in cat_paper.articles:
                try:
                    print("====",article.url)
                    add_news(conn,article.url,cat_id)
                except Exception as ex:
                    print("ERROR:" + str(ex))
                    pass

        conn.close()
```

**search_by_keywords(keywords = None)**

```
#define function search theo keyword truyền vào form
def search_by_keywords(keywords = None):
    conn = sqlite3.connect("DATA/data.db")
    if keywords is not None:
        # news = get_all("SELECT * FROM news")
        search_news_by_kw = [n for n in get_all("SELECT * FROM news") if n[1].lower().find(keywords.lower()) >= 0]
        data = []
        for n in search_news_by_kw:
            data.append(get_news_id(n[0]))
        # print(data)
    conn.close()
    return data
```

- Define các hàm trong api

```
#render templates
@app.route("/")
def render():
    return render_template("index.html")
```

```
# get category in database
@app.route("/category", methods = ["GET"]) # define endpoint để thực thi API
def get_categories():
    rows = utils.get_all("SELECT  * FROM category")
    data = []
    for r in rows:
        data.append(
            { "id": r[0],
              "subject": r[1],
              "url": r[2]
            }
        )
    #return jsonify({"category": data})
    return render_template("category.html", data=data)
```

```
# get tất cả news from database
@app.route("/news", methods = ["GET"]) # define endpoint để thực thi API
def get_news():
    kw = request.args.get("keywords", None)
    return render_template("news.html", data=render_templates.read_news(kw))
"""
```

```python
# 2. API get news from database
@app.route("/news", methods = ["GET"]) # define endpoint để thực thi API
def get_news():
    rows = utils.get_all("SELECT * FROM news")
    data = []
    for r in rows:
        data.append(
        {
            "id": r[0],
            "subject": r[1],
            "description": r[2],
            "image": r[3],
            "original_url": r[4],
            "category_id": r[5]
        })
    return jsonify({"news": data})
```

**Tạo module mới có tên render_templates.py, sau đó tạo các function sau:**

**read_category()**

```python
#ead category
def read_category():
    rows = utils.get_all("SELECT  * FROM category")
    data = []
    for r in rows:
        data.append(
            {"id": r[0],
             "subject": r[1],
             "url": r[2]
             }
        )
    # with open("json_file/category.json", "w") as cat:
    #    json.dump(data, cat)
    return render_template("category.html", data=data)
```

**get_news()**

```
def get_news():
    rows = utils.get_all("SELECT * FROM news")
    data = []
    for r in rows:
        data.append(
        {
            "id": r[0],
            "subject": r[1],
            "description": r[2],
            "image": r[3],
            "original_url": r[4],
            "category_id": r[5]
        })
    with open("json_file/news.json", "w", encoding="utf8") as f:
        json.dump(data, f)
```

**read_news()**

```
def read_news(keywords = None):
    with open("json_file/news.json", encoding="utf8") as f:
        news = json.load(f)
    if keywords:
        news = [n for n in news if n["subject"].lower().find(keywords.lower()) >= 0]
    return news
```

**Thư mục templates**

Tạo thư mục templates, sau đó tạo các file category.html (chứa danh mục các trang chứa dữ liệu cần thu thập) , news.html (chứa dữ liệu các bài báo thu thập được) và index.html (trang chủ của webstie).

Cấu trúc các file html như sau:

**Category.html:**

```
<body>
<h1> DANH MỤC TRANG  TÌM KIẾM</h1>
    <table class ="table">
        <tr>
            <th> ID </th>
            <th> Tên trang báo </th>
            <th> Link trang chủ bài báo</th>
        </tr>
        {% for cat in data %}
        <tr>
            <td>
                {{cat.id}}
            </td>
            <td>
                {{cat.subject}}
            </td>
            <td>
                {{cat.url}}
            </td>
        </tr>
        {% endfor %}
    </table>
```

**News.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title> DANH MỤC BÀI BÁO </title>
</head>
<body>
<h1> DANH MỤC CÁC BÀI BÁO </h1>
<p>
    <form>
        <div class = "form-group">
            <label> Tìm kiếm theo tiêu đề </label>
            <input type = "text" name = "keywords"> <br>
            <input type = "submit" value = "Tìm">
        </div>
    </form>
</p>
```

```
<table class ="table">
        <tr>
            <th> ID </th>
            <th> Tiêu đề </th>
            <th> Link gốc bài </th>
            <th> ID danh mục </th>
        </tr>
        {% for new in data %}
        <tr>
            <td>
                {{new.id}}
            </td>
            <td>
                {{new.subject}}
            </td>
            <td>
                {{new.original_url}}
            </td>
            <td>
                {{new.category_id}}
            </td>
        </tr>
        {% endfor %}
    </table>
```

**Index.html**

```
<h1>TÌM KIẾM BÀI BÁO</h1>
<p>
    <form action = "/news/<keywords>">
        <div class = "form-group">
            <label> Tìm theo keywords</label>
            <input type = "text" name = "keywords"> <br>
            <input type = "submit" value = "Tìm">
        </div>

    </form>
</p>
```

Sau khi hoàn thành việc lấy dữ liệu và render lên website, thực hiện gắn CSS cho website.