BÀI THỰC HÀNH SỐ 5

Lập trình giao diện và kết nối cơ sở dữ liệu – Phần 2

I. Mục tiêu

- Nắm được các thư viện hỗ trợ phát triển ứng dụng có giao diện
- Tìm hiểu về cách sử dụng Tkinter để xây dựng giao diện
- Xây dựng các ứng dụng có giao diện đơn giản
- Nắm được các thư viện hỗ trợ kết nối cơ sở dữ liệu
- Biết cách thực thi các lệnh truy vấn dữ liệu từ cơ sở dữ liệu
    - Thực thi các lệnh truy vấn SELECT, INSERT, UPDATE, DELETE
    - Thực thi các lệnh truy vấn dùng tham số
    - Thực thi thủ tục

II. Hướng dẫn và bài tập

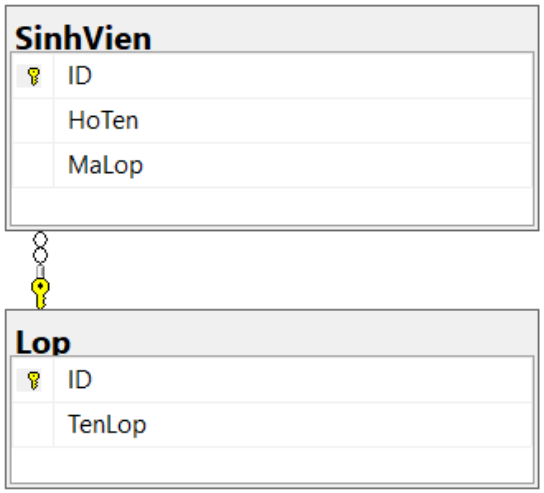
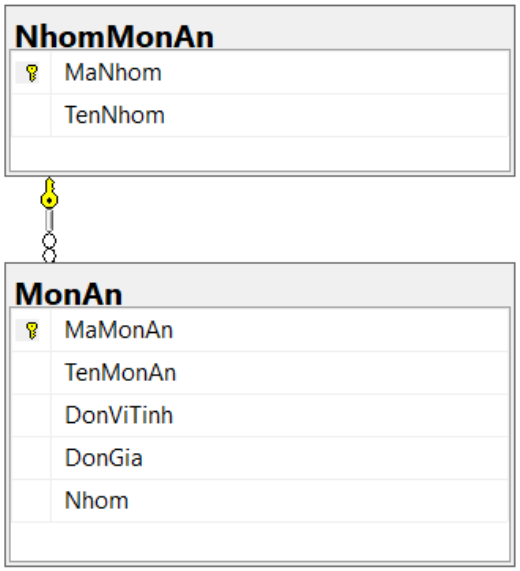
Để kết nối với các hệ quản trị cơ sở dữ liệu khác nhau, chúng ta cần cài đặt các thư viện hỗ trợ kết nối. Mỗi hệ quản trị cơ sở dữ liệu cũng có một hoặc nhiều thư viện kết nối. Bài thực hành này giới thiệu một số thư viện hỗ trợ kết nối tới các hệ quản trị cơ sở dữ liệu phổ biến như SQL server, MySQL, PostgreSQL và SQLite. Nếu máy tính sinh viên chưa cài hệ quản trị CSDL nào, sinh viên có thể cài đặt SQLite để thực hành các bài lab này.

SQLite là hệ quả trị cơ sở dữ liệu (DBMS) quan hệ tương tự như Mysql, ... Đặc điểm nổi bật của SQLite so với các DBMS khác là gọn, nhẹ, đơn giản, đặt biệt không cần mô hình server-client, không cần cài đặt, cấu hình hay khởi động nên không có khái niệm user, password hay quyền hạn trong SQLite Database. Dữ liệu cũng được lưu ở một file duy nhất (Đọc thêm ở : https://hoc.tv/hoc-sqlite)

Một số thư viện hỗ trợ kết nối cơ sở dữ liệu dùng Python được liệt kê dưới đây. Tùy thuộc vào máy tính của bạn đang cài đặt sẵn hệ quản trị CSDL nào thì bạn cài thư viện tương ứng:

- MySQL: mysql connector python (https://boxhoidap.com/huong-dan-pip-install-mysql-connector)
- SQL server: pyodbc (https://learn.microsoft.com/vi-vn/sql/connect/python/pyodbc/step-1-configure-development-environment-for-pyodbc-python-development?view=sql-server-2017 0
- PostgreSQL: psycopg2 (https://hocdevops.com/python/lam-viec-voi-postgresql-trong-python-su-dung-psycopg2/)
- SQLite: sqlite3 (https://hoc.tv/hoc-sqlite/cai-dat-sqlite-1961)

Cơ sở dữ liệu mẫu dùng cho bài thực hành: bài thực hành này dùng 2 CSDL có sô đồ quan hệ như dưới đây. Bạn có thể tùy chỉnh (kiểu dữ liệu của các trường) lệnh SQL trong 2 file QLSinhVien.sql và QLMonAn.sql để tạo CSDL trong hệ quản trị CSDL bạn đang sử dụng

CSDL QLSinhVien	CSDL QLMonAn
	

Bài tập 1: Thực thi lệnh truy vấn SELECT và các lệnh CRUD đơn giản

Ghi chú: Bài này dùng cơ sở dữ liệu QLSinhVien

1. ***Kết nối tới cơ sở dữ liệu và lấy thông tin phiên bản***

Trình tự cơ bản gồm 3 bước như sau:

- Viết truy vấn SQL để lấy phiên bản cơ sở dữ liệu
- Kết nối tới cơ sở dữ liệu và sử dụng ***cursor.excute()*** để thực thi câu truy vấn
- Tiếp theo, dùng ***cursor.fetchone()*** để nạp (lấy) dữ liệu

Đoạn mã sau đây minh họa cách đơn giản nhất kết nối CSDL và lấy phiên bản hệ quản trị CSDL SQL server

```
import pyodbc

conn = pyodbc.connect('DRIVER={ODBC Driver 18 for SQL Server};SERVER=.;DATABASE=QLMonAn;UID=sa;PWD=sa;Encrypt=no')
cursor = conn.cursor()
cursor.execute("SELECT @@version")
conn.close()
db_version = cursor.fetchone()
print("Bạn đang dùng hệ quản trị CSDL SQL server phiên bản ", db_version)
```

Đoạn mã sau đây minh họa cách lấy phiên bản hệ quản trị CSDL SQLite, đã tổ chức thành các hàm kết nối và đóng kết nối riêng biệt, có sử dụng try… exception … để xử lý nếu có lỗi xảy ra khi kết nối và thực thi truy vấn

```python
import sqlite3

def get_connection():
    connection = sqlite3.connect('QLSinhVien.db')
    return connection

def close_connection(connection):
    if connection:
        connection.close()

def read_database_version():
    try:
        connection = get_connection()
        cursor = connection.cursor()
        cursor.execute("select sqlite_version();")
        db_version = cursor.fetchone()
        print("Bạn đang sử dụng SQLite phiên bản: ", db_version)
        close_connection(connection)
    except (Exception, sqlite3.Error) as error:
        print("Đã có lỗi xảy ra. Thông tin lỗi: ", error)

read_database_version()
```

Với hệ quản trị CSDL khác, bạn có thể import thư viên, thay đổi chuỗi kết nối và sử dụng lệnh lấy phiên bản cho phù hợp còn quá trình thực thi lệnh gần như tương tự.

2. ***Lấy danh sách của lớp học, sinh viên:*** đoạn mã minh họa kết nối với cơ sở dữ liệu SQL Server dùng pyodbc

```python
import pyodbc

connectionString = '''DRIVER={ODBC Driver 18 for SQL Server};
                      SERVER=.;DATABASE=QLSinhVien;UID=sa;PWD=sa;Encrypt=no'''

def get_connection():
    conn = pyodbc.connect(connectionString)
    return conn

def close_connection(conn):
    if conn:
        conn.close()

def get_all_class():
    try:
        connection = get_connection()
        cursor = connection.cursor()

        select_query = """select * from Lop"""
        cursor.execute(select_query)
```

```python
        records = cursor.fetchall()

        print(f"Danh sách các lớp là: ")
        for row in records:
            print("*"*50)
            print("Mã lớp: ", row[0])
            print("Tên lớp: ", row[1])

        close_connection(connection)
    except (Exception, pyodbc.Error) as error:
        print("Đã có lỗi xảy ra khi thực thi. Thông tin lỗi: ", error)


get_all_class()
```

Kết quả hiển thị

```
Danh sách các lớp là:
**************************************************
Mã lớp:  1
Tên lớp:  CTK43
**************************************************
Mã lớp:  2
Tên lớp:  CTK44A
**************************************************
Mã lớp:  3
Tên lớp:  CTK44B
**************************************************
Mã lớp:  4
Tên lớp:  CTK45A
```

Tương tự bạn hãy viết hàm đọc thông tin sinh viên để kết quả hiển thị như sau:

```
Danh sách tất cả sinh viên là:
Mã số       Họ tên                                  Mã lớp
1       Trần Văn Thái                               1
2       Mai Thành Thân                              2
3       Phạm Thanh Thảo                             2
4       Trần Quốc Bảo Trung                         3
5       Thái Thành Lam                              3
6       Trần Văn Tám                                3
7       Nguyễn Công Thành                           4
8       Nguyễn Thị Lụa                              1
9       Phan Thanh Nga                              1
10      Trương Công Quyền                           4
11      Võ Thị Sáu                                  1
12      Võ Tòng                                     2
```

Chỉnh sửa câu truy vấn để đọc lên tên lớp, kết quả như hình sau:

```
Danh sách tất cả sinh viên là:
Mã số       Họ tên                  Mã lớp      Tên lớp
1       Trần Văn Thái               1           CTK43
2       Mai Thành Thân              2           CTK44A
3       Phạm Thanh Thảo             2           CTK44A
4       Trần Quốc Bảo Trung         3           CTK44B
5       Thái Thành Lam              3           CTK44B
6       Trần Văn Tám                3           CTK44B
7       Nguyễn Công Thành           4           CTK45A
8       Nguyễn Thị Lụa              1           CTK43
9       Phan Thanh Nga              1           CTK43
10      Trương Công Quyền           4           CTK45A
11      Võ Thị Sáu                  1           CTK43
12      Võ Tòng                     2           CTK44A
13      Abc sds                     4           CTK45A
```

3. ***Lấy thông tin lớp học, sinh viên theo mã:*** bổ sung hàm sau vào đoạn chương trình

```python
def get_class_by_id(class_id):
    try:
        connection = get_connection()
        cursor = connection.cursor()

        # dấu "?" chính là placeholder, điểm đánh dấu vị trí tham số được truyền vào
        select_query = "select * from Lop where id = ?"
        # danh sách tham số sẽ truyền vào câu truy vấn
        params = (class_id,)
        cursor.execute(select_query, params)

        record = cursor.fetchone()

        print(f"Thông tin lớp có id = {class_id} là: ")
        print("Mã lớp: ", record[0])
        print("Tên lớp: ", record[1])

        close_connection(connection)
    except (Exception, pyodbc.Error) as error:
        print("Đã có lỗi xảy ra khi thực thi. Thông tin lỗi: ", error)

get_class_by_id(1)
```

Lưu ý: Đối với thư viện pyodbc, "?" được dùng làm điểm đánh dấu tham số (placeholder), còn đối với sqlite3 và mysqlconnector thì "%s" sẽ được sử dụng

Tương tự, bạn hãy hoàn thành các yêu cầu sau:

- Viết hàm hiển thị thông tin sinh viên theo mã sinh viên
- Hiển thị danh sách sinh viên theo lớp (khi biết mã lớp/tên lớp)
- Tìm kiếm thông tin sinh viên theo tên và lớp, ví dụ như hình sau

```
110
111    find_student(3,"Trung")
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

Danh sách tất cả sinh viên tên Trung ở lớp có mã 3:
Mã số       Họ tên                  Mã lớp
4        Trần Quốc Bảo Trung           3
```

4. ***Insert/Update/Delete:*** bổ sung hàm sau vào chương trình

```python
def insert_class(class_name):
    try:
        connection = get_connection()
        cursor = connection.cursor()


        # Cách 1 - truyền trực tiếp tham số vào câu truy vấn
        #select_query = f"Insert into Lop(TenLop) values ('{class_name}')"
        #cursor.execute(select_query)

        # Cách 2 - Dùng tham số
        select_query = "Insert into Lop(TenLop) values ( ? )"
        cursor.execute(select_query, (class_name,))

        connection.commit()


        print("Đã thêm thành công")

        close_connection(connection)
    except (Exception, pyodbc.Error) as error:
        print("Đã có lỗi xảy ra khi thực thi. Thông tin lỗi: ", error)
```

Gọi hàm và kiểm tra kết quả

```
134    insert_class("CTK44AB")
135    get_all_class()
```

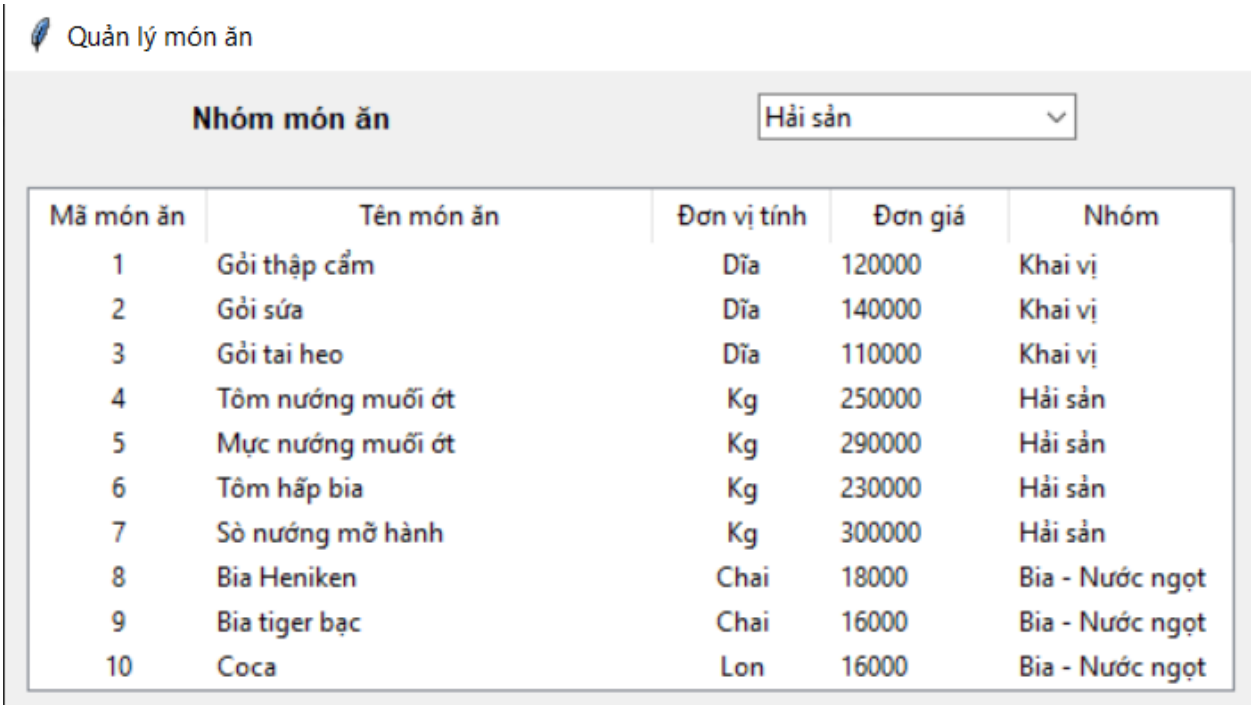
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

Đã thêm thành công
Danh sách các lớp là:
**************************************************
Mã lớp:  1
Tên lớp:  CTK43
**************************************************
Mã lớp:  2
Tên lớp:  CTK44A
**************************************************
Mã lớp:  3
Tên lớp:  CTK44B
**************************************************
Mã lớp:  4
Tên lớp:  CTK45A
**************************************************
Mã lớp:  1003
Tên lớp:  CTK44AB
```

Hãy viết hàm thêm, sửa, xóa thông tin sinh viên (có thể tìm hiểu cách gọi thủ tục)

Bài tập 2: Xây dựng form quản lý thông tin món ăn

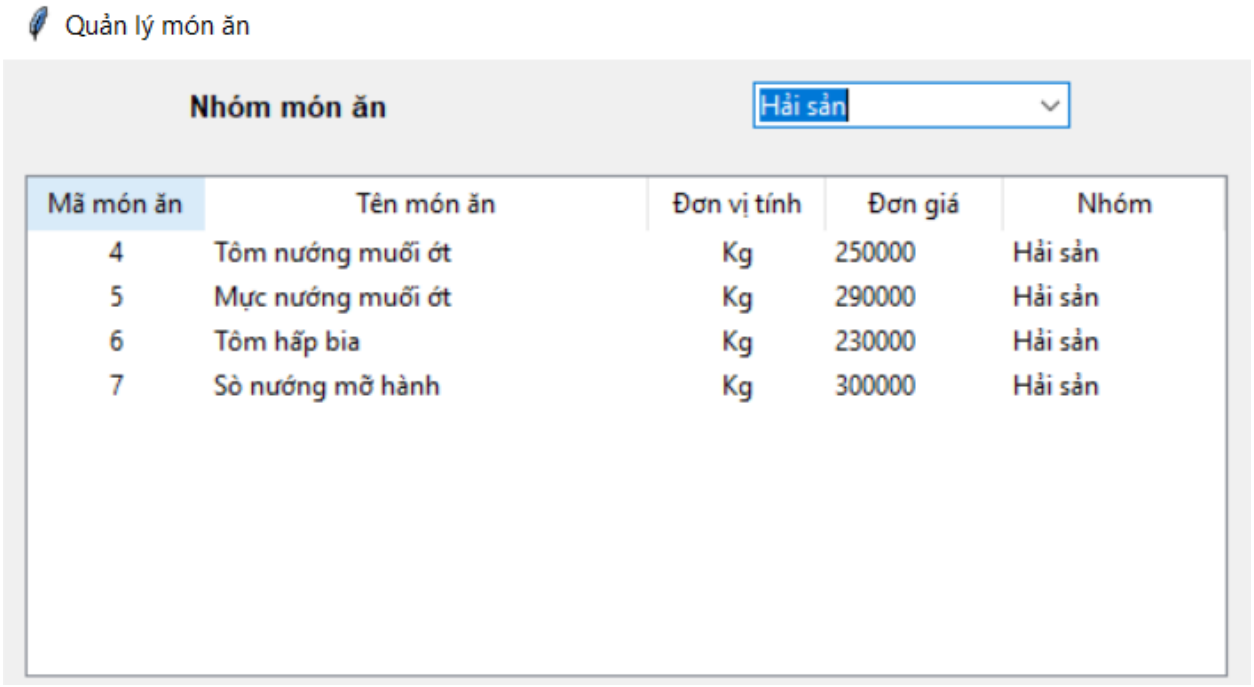
Lưu ý: Bài tập này sử dụng cơ sở dữ liệu ***QLMonAn***

1. Thiết kế form Quản lý món ăn như hình dưới đây



Mã món ăn	Tên món ăn	Đơn vị tính	Đơn giá	Nhóm
1	Gỏi thập cẩm	Dĩa	120000	Khai vị
2	Gỏi sứa	Dĩa	140000	Khai vị
3	Gỏi tai heo	Dĩa	110000	Khai vị
4	Tôm nướng muối ớt	Kg	250000	Hải sản
5	Mực nướng muối ớt	Kg	290000	Hải sản
6	Tôm hấp bia	Kg	230000	Hải sản
7	Sò nướng mỡ hành	Kg	300000	Hải sản
8	Bia Heniken	Chai	18000	Bia - Nước ngọt
9	Bia tiger bạc	Chai	16000	Bia - Nước ngọt
10	Coca	Lon	16000	Bia - Nước ngọt

2. Khi chọn nhóm món ăn bất kỳ, danh sách hiển thị các món ăn thuộc nhóm như sau:



Mã món ăn	Tên món ăn	Đơn vị tính	Đơn giá	Nhóm
4	Tôm nướng muối ớt	Kg	250000	Hải sản
5	Mực nướng muối ớt	Kg	290000	Hải sản
6	Tôm hấp bia	Kg	230000	Hải sản
7	Sò nướng mỡ hành	Kg	300000	Hải sản

3. Bổ sung thêm chức năng thêm, xóa, sửa món ăn