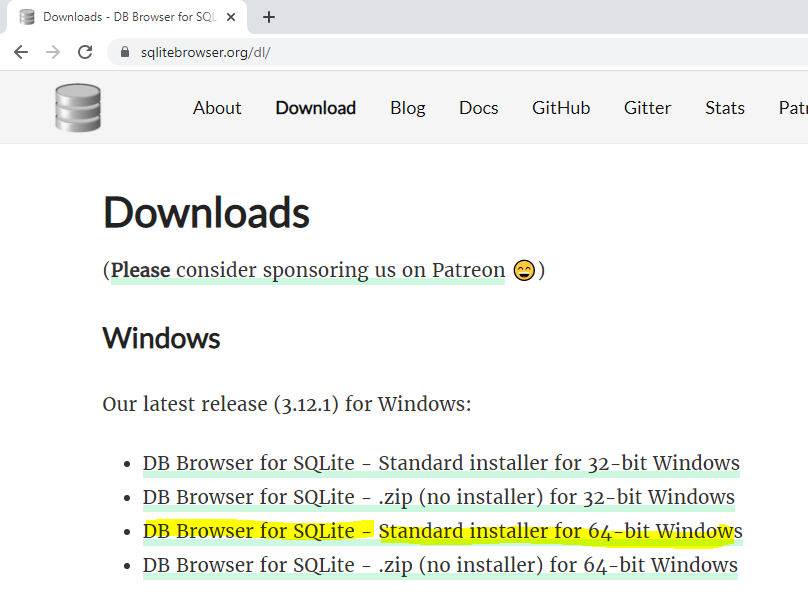
# Cài đặt trình soạn thảo database SQLite

Vào trang web <https://sqlitebrowser.org/> để download phần mềm.

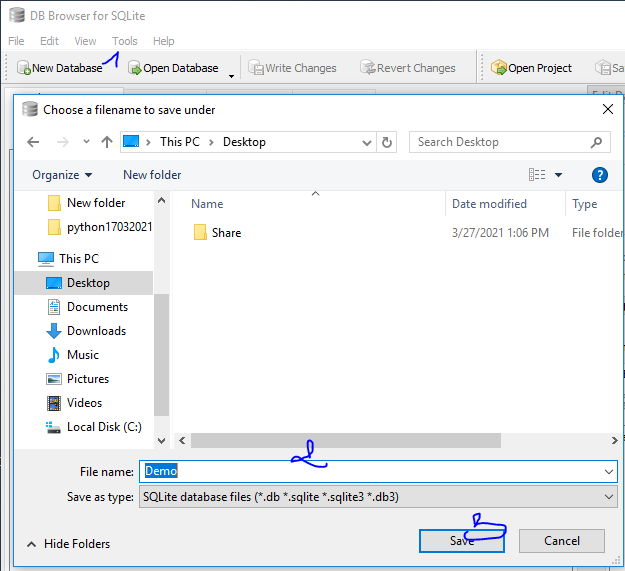


Bạn có thể sử dụng bản cài đặt hoặc bản Zip không cần cài đặt.

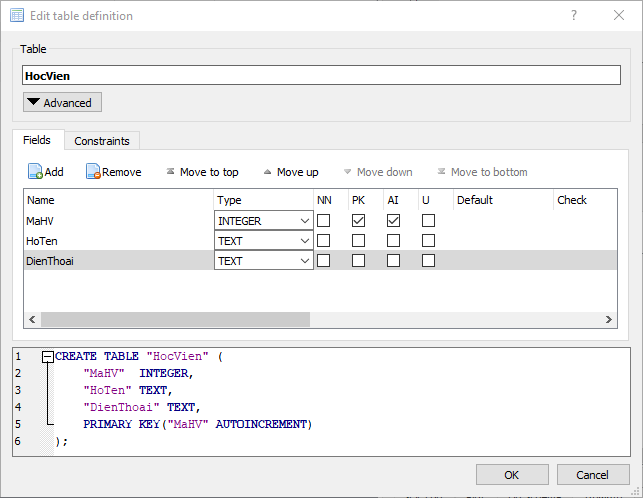
# Thao tác trên database

## Tạo mới Database

Bấm button **New Database** và điền tên file database.



## Tạo bảng HocVien



CREATE TABLE "HocVien" (

"MaHV" INTEGER,

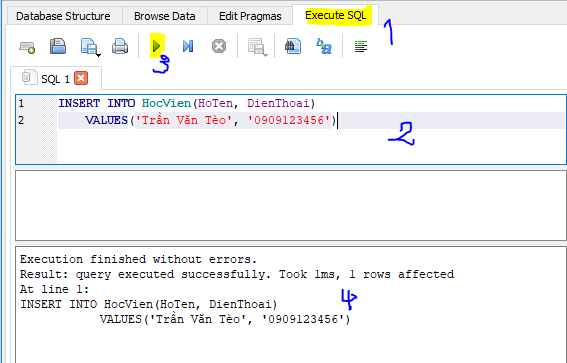
"HoTen" TEXT,

"DienThoai" TEXT,

PRIMARY KEY("MaHV" AUTOINCREMENT)

);

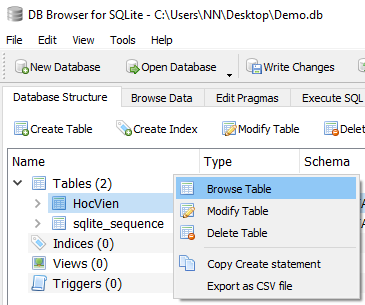
## Nhập liệu bằng dòng lệnh:

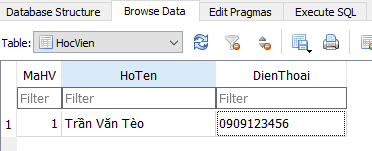


INSERT INTO HocVien(HoTen, DienThoai)

VALUES('Nguyễn Văn Tèo', '0909123456');

## Xem data trong bảng:





## Thực hiện thao tác bằng câu lệnh SQL

# Thao tác với SQLite3

## Lấy thông tin database

import sqlite3  
  
try:  
 sqlite\_Connection = sqlite3.connect(**'PythonDB.db'**)  
 conn = sqlite\_Connection.cursor()  
 print(**"**\n**Database created and connected to SQLite."**)  
 sqlite\_select\_Query = **"select sqlite\_version();"** conn.execute(sqlite\_select\_Query)  
 record = conn.fetchall()  
 print(**"**\n**SQLite Database Version is: "**, record)  
 conn.close()  
except sqlite3.Error as error:  
 print(**"**\n**Error while connecting to sqlite"**, error)  
finally:  
 if (sqlite\_Connection):  
 sqlite\_Connection.close()  
 print(**"**\n**The SQLite connection is closed."**)

## Thực hiện tạo bảng

import sqlite3  
from sqlite3 import Error  
  
  
def sql\_connection():  
 try:  
 conn = sqlite3.connect(**'PythonDB.db'**)  
 return conn  
 except Error:  
 print(Error)  
  
  
def sql\_table(conn):  
 cursorObj = conn.cursor()  
 cursorObj.execute(  
 **"CREATE TABLE HangHoa(MaHH char(6) PRIMARY KEY, TenHH char(40), MoTa char(55),DonGia decimal(10,2),SKU char(15) NULL);"**)  
 print(**"**\n**Đã tạo bảng HangHoa."**)  
 conn.commit()  
  
  
sqllite\_conn = sql\_connection()  
sql\_table(sqllite\_conn)  
if (sqllite\_conn):  
 sqllite\_conn.close()  
 print(**"**\n**Đóng kết nối."**)

## Thực hiện thêm data cho bảng HocVien

# Quản lý Hàng Hóa

## Tạo table Loai & HangHoa

import sqlite3  
from sqlite3 import Error

def create\_connection(db\_file):  
 *""" create a database connection to the SQLite database specified by db\_file.  
 :param db\_file: database file  
 :return: Connection object or None  
 """* connection = None  
 try:  
 connection = sqlite3.connect(db\_file)  
 except Error as e:  
 print(e)  
  
 return connection  
  
  
def create\_table(connection, create\_table\_sql):  
 *""" create a table from the create\_table\_sql statement  
 :param connection: Connection object  
 :param create\_table\_sql: a CREATE TABLE statement  
 :return:  
 """* try:  
 c = connection.cursor()  
 c.execute(create\_table\_sql)  
 except Error as e:  
 print(e)

def main():  
 database = **r"pythonsqlite.db"** sql\_create\_table\_loai = **"""  
 CREATE TABLE Loai(  
 MaLoai integer PRIMARY KEY,  
 TenLoai varchar(40)  
 )   
 """** sql\_create\_table\_hang\_hoa = **"""  
 CREATE TABLE HangHoa(  
 MaHH char(6) PRIMARY KEY,  
 TenHH varchar(40),   
 MoTa varchar(55),  
 DonGia decimal(10,2),  
 SKU varchar(15) NULL,  
 MaLoai interger NULL,  
 FOREIGN KEY (MaLoai) REFERENCES Loai(MaLoai)  
 )   
 """** *# create a database connection* conn = create\_connection(database)  
  
 *# create tables* if conn is not None:  
 *# create Loai table* create\_table(conn, sql\_create\_table\_loai)  
  
 *# create HangHoa table* create\_table(conn, sql\_create\_table\_hang\_hoa)  
  
 *# close connection* conn.close()  
 else:  
 print(**"Error! cannot create the database connection."**)  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

## Chèn dữ liệu vào bảng Loai, HangHoa

import sqlite3  
from sqlite3 import Error  
import dbcommon  
  
  
def create\_loai\_sql\_statement(conn, ten\_loai):  
 *"""Create loai data."""* sql = **f''' INSERT INTO Loai(TenLoai) VALUES('**{ten\_loai}**') '''** cur = conn.cursor()  
 cur.execute(sql)  
 conn.commit()  
 return cur.lastrowid  
  
  
def create\_hang\_hoa(conn, hang\_hoa\_obj):  
 *"""Create a hanghoa."""* sql = **''' INSERT INTO HangHoa(MaHH,TenHH,MoTa,DonGia,SKU,MaLoai)  
 VALUES(?,?,?,?,?,?) '''** cur = conn.cursor()  
 cur.execute(sql, hang\_hoa\_obj)  
 conn.commit()  
 return cur.lastrowid  
  
  
def main():  
 database = **r"pythonsqlite.db"** *# create a database connection* conn = dbcommon.create\_connection(database)  
 with conn:  
 *# Tao loai* loai\_dien\_tu\_id = create\_loai\_sql\_statement(conn, **'Điện tử'**)  
 loai\_dien\_lanh\_id = create\_loai\_sql\_statement(conn, **'Điện lạnh'**)  
  
 *# Tạo hàng hóa - (MaHH,TenHH,MoTa,DonGia,SKU,MaLoai)* hang\_hoa\_1 = (**'SP0001'**, **'Monitor DELL'**, **'19inc, Full HD'**, 3990, **'SKU0001'**, loai\_dien\_tu\_id)  
 hang\_hoa\_2 = (**'SP0002'**, **'Dailin 9912'**, **'Máy lạnh Daikin 9923, 1HP, 2021'**, 9990, **'SKU0002'**, loai\_dien\_lanh\_id)  
 create\_hang\_hoa(conn, hang\_hoa\_1)  
 create\_hang\_hoa(conn, hang\_hoa\_2)  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

## Cập nhật dữ liệu

import sqlite3  
from sqlite3 import Error  
import dbcommon  
  
def update\_hanghoa(conn, product):sql = **''' UPDATE HangHoa  
 SET TenHH = ? ,  
 MoTa = ? ,  
 DonGia = ?  
 WHERE MaHH = ?'''** cur = conn.cursor()  
 cur.execute(sql, task)  
 conn.commit()  
  
  
def main():  
 database = **r"pythonsqlite.db"** *# create a database connection* conn = dbcommon.create\_connection(database)  
 with conn:  
 update\_hanghoa(conn, (**'Beer SaiGon'**, **'Beer SaiGon'**, 19500, 2))  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

## Truy vấn lấy dữ liệu

import sqlite3  
from sqlite3 import Error  
import dbcommon  
  
  
def select\_all\_loai(conn):  
 *"""  
 Query all rows in the Loai table  
 :param conn: the Connection object  
 :return:  
 """* cur = conn.cursor()  
 cur.execute(**"SELECT \* FROM Loai"**)  
  
 rows = cur.fetchall()  
  
 for row in rows:  
 print(row)  
  
  
def select\_all\_data(conn, sql\_statement):  
 *"""  
 Query all rows in the Loai table  
 :param sql\_statement: the sql statement  
 :type sql\_statement: str  
 :param conn: the Connection object  
 :return:  
 """* cur = conn.cursor()  
 cur.execute(sql\_statement)  
  
 rows = cur.fetchall()  
  
 for row in rows:  
 print(row)  
  
  
def main():  
 database = **r"pythonsqlite.db"** *# create a database connection* conn = dbcommon.create\_connection(database)  
 with conn:  
 print(**"1. Query all Loais"**)  
 select\_all\_loai(conn)  
  
 print(**"2. Query all HangHoas:"**)  
 select\_all\_data(conn, **"SELECT \* FROM HangHoa"**);  
  
 print(**"3. Query all HangHoas by Loai:"**)  
 sql\_get\_products\_by\_category = **"SELECT \* FROM HangHoa WHERE MaLoai = 2"** select\_all\_data(conn, sql\_get\_products\_by\_category);  
  
 sql\_get\_products\_by\_category = **"""  
 SELECT \* FROM HangHoa as HH JOIN Loai as lo ON lo.MaLoai = HH.MaLoai  
 WHERE TenLoai = 'Điện tử'  
 """** select\_all\_data(conn, sql\_get\_products\_by\_category);  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

## Xóa data

import sqlite3  
from sqlite3 import Error  
import dbcommon  
  
  
def delete\_hanghoa\_by\_id(conn, id):  
 *"""  
 Delete a HangHoa by id  
 :param conn: Connection to the SQLite database  
 :param id: id of the task  
 :return:  
 """* sql = **'DELETE FROM HangHoa WHERE MaHH=?'** cur = conn.cursor()  
 cur.execute(sql, (id,))  
 conn.commit()  
  
  
def delete\_all\_hang\_hoa(conn):  
 *"""  
 Delete all rows in the HangHoa table  
 :param conn: Connection to the SQLite database  
 :return:  
 """* sql = **'DELETE FROM HangHoa'** cur = conn.cursor()  
 cur.execute(sql)  
 conn.commit()  
  
  
def main():  
 database = **r"pythonsqlite.db"** *# create a database connection* conn = dbcommon.create\_connection(database)  
 with conn:  
 delete\_hanghoa\_by\_id(conn, 2);  
 *# delete\_all\_hang\_hoa(conn);*if \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

# Quản lý Sách

CREATE TABLE TacGia (

MaTacGia INTEGER NOT NULL PRIMARY KEY,

Ho VARCHAR,

Ten VARCHAR

);

CREATE TABLE Sach (

MaSach INTEGER NOT NULL PRIMARY KEY,

MaTacGia INTEGER REFERENCES author,

TenSach VARCHAR

);

CREATE TABLE NhaXB (

MaNXB INTEGER NOT NULL PRIMARY KEY,

TenNXB VARCHAR

);

