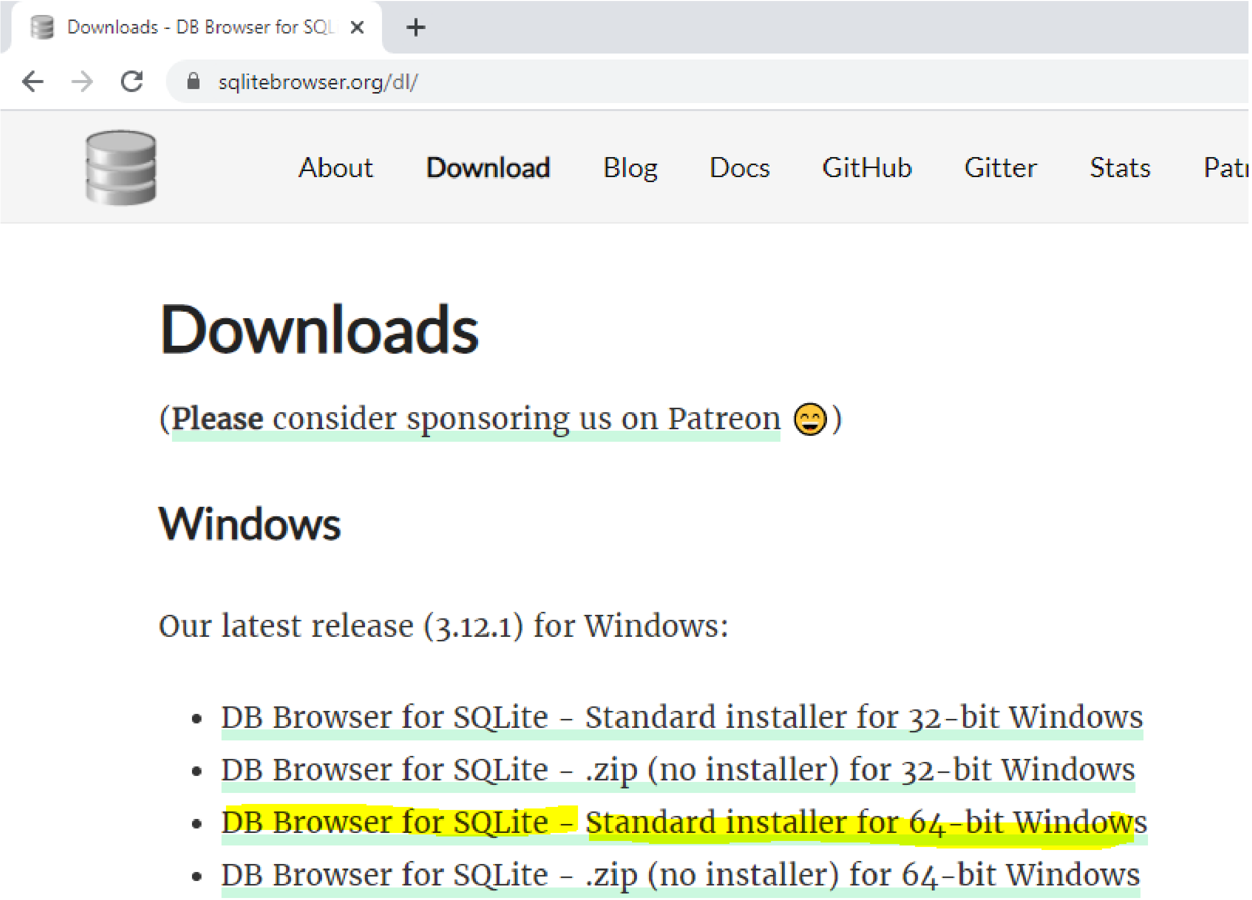
1 Cài đặt trình soạn thảo database SQLite

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



Downloads

(Please consider sponsoring us on Patreon 😄)

Windows

Our latest release (3.12.1) for Windows:

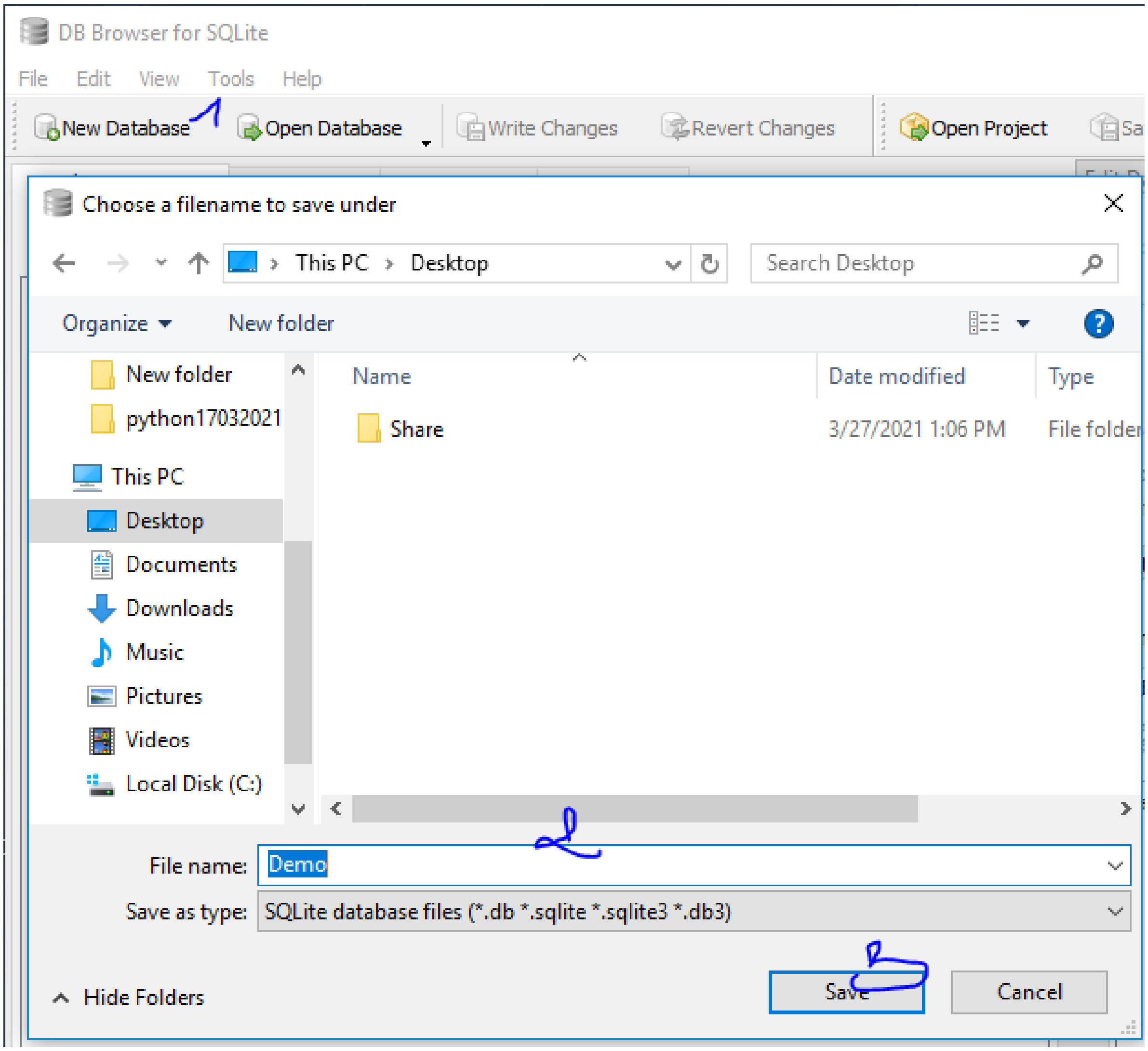
- DB Browser for SQLite - Standard installer for 32-bit Windows
- DB Browser for SQLite - .zip (no installer) for 32-bit Windows
- DB Browser for SQLite - Standard installer for 64-bit Windows
- DB Browser for SQLite - .zip (no installer) for 64-bit Windows

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

2 Thao tác trên database

2.1 Tạo mới Database

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



Name	Date modified	Type
Share	3/27/2021 1:06 PM	File folder

File name: Demo

Save as type: SQLite database files (*.db *.sqlite *.sqlite3 *.db3)

2.2 Tạo bảng HocVien

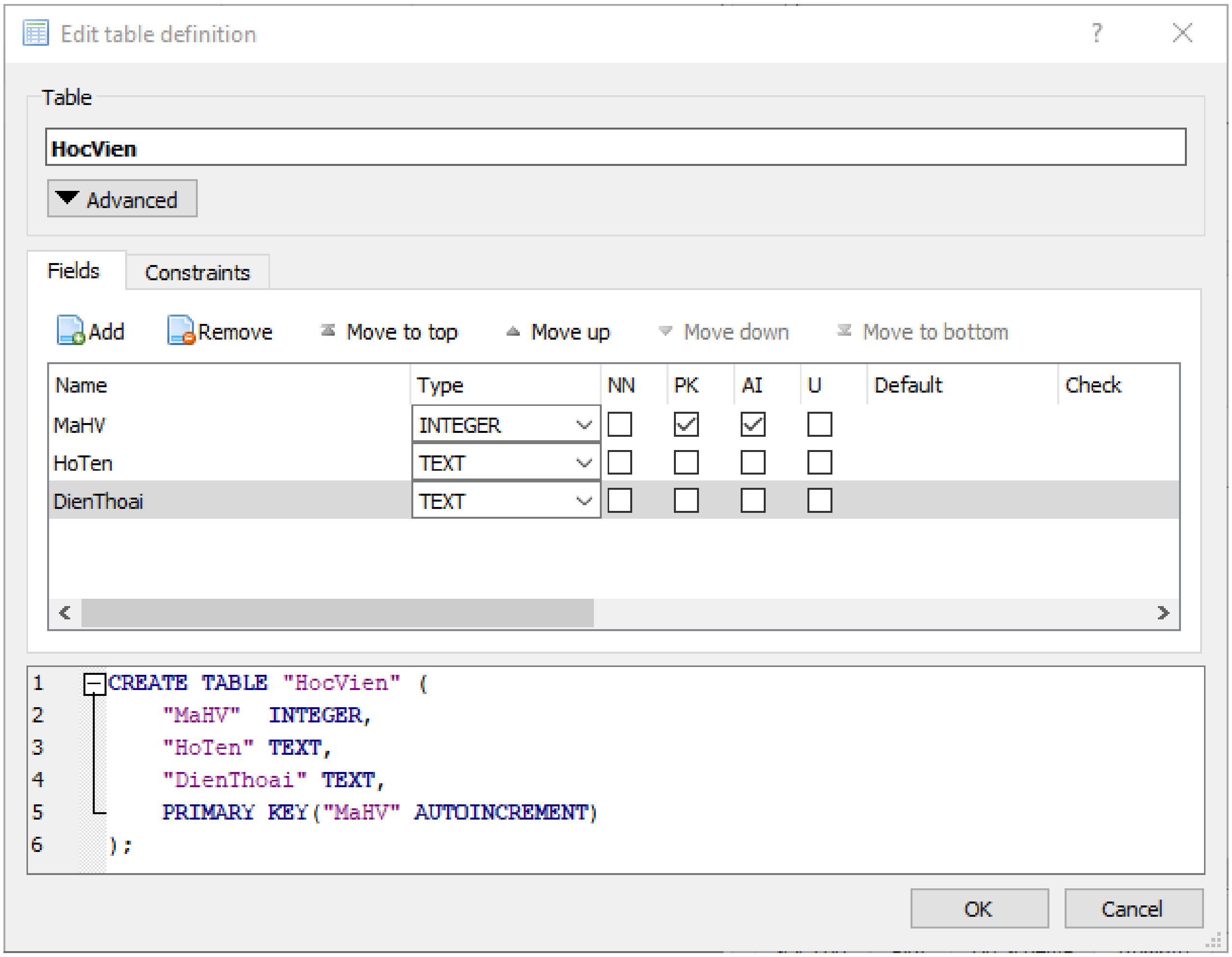


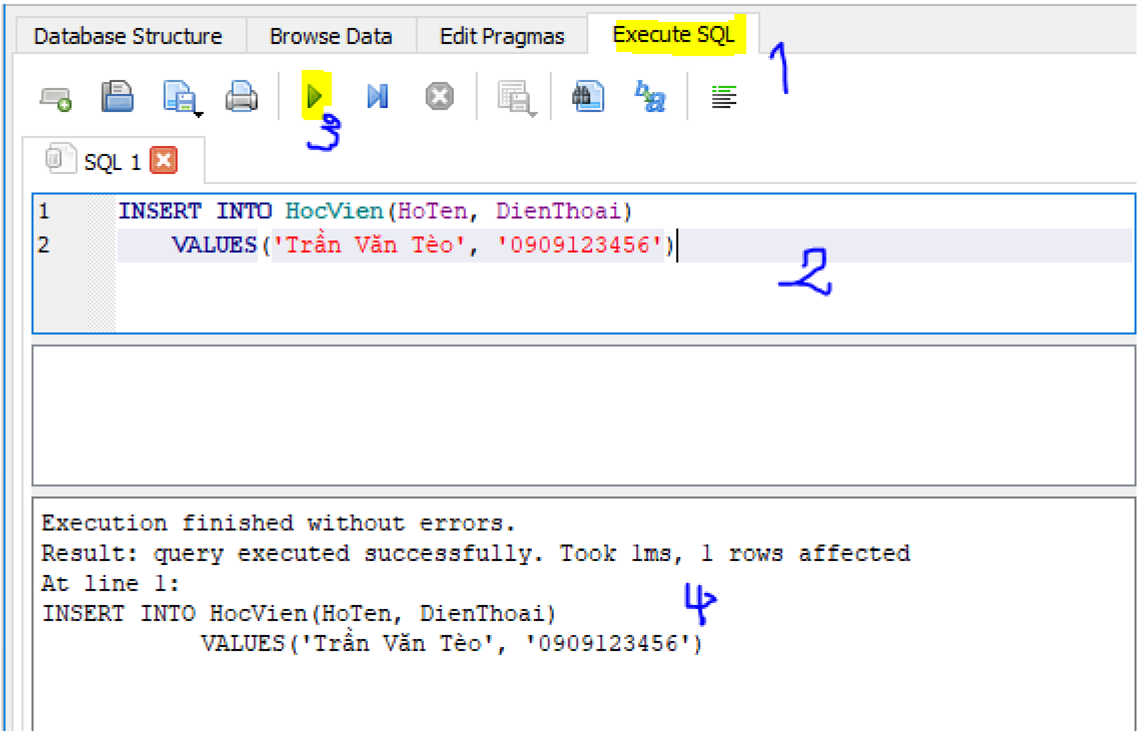
Table: HocVien

Name	Type	NN	PK	AI	U	Default	Check
MaHV	INTEGER	☐	☑	☑	☐		
HoTen	TEXT	☐	☐	☐	☐		
DienThoai	TEXT	☐	☐	☐	☐		

```
CREATE TABLE "HocVien" (
	"MaHV"	INTEGER,
	"HoTen"	TEXT,
	"DienThoai"	TEXT,
	PRIMARY KEY("MaHV" AUTOINCREMENT)
);
```

```
CREATE TABLE "HocVien" (
	"MaHV"		INTEGER,
	"HoTen"		TEXT,
	"DienThoai"	TEXT,
	PRIMARY KEY("MaHV" AUTOINCREMENT)
);
```

2.3 Nhập liệu bằng dòng lệnh:



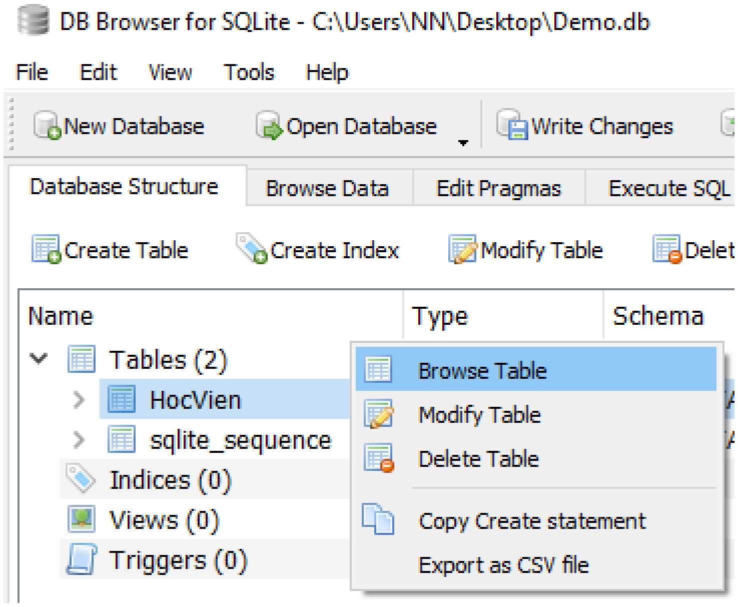
```
INSERT INTO HocVien(HoTen, DienThoai)
    VALUES('Trần Văn Tèo', '0909123456')
```

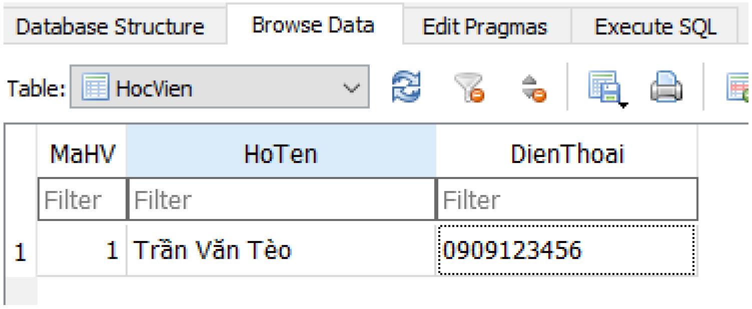
```
Execution finished without errors.
Result: query executed successfully. Took 1ms, 1 rows affected
At line 1:
INSERT INTO HocVien(HoTen, DienThoai)
           VALUES('Trần Văn Tèo', '0909123456')
```

INSERT INTO HocVien(HoTen, DienThoai)

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

2.4 Xem data trong bảng:





	MaHV	HoTen	DienThoai
1	1	Trần Văn Tèo	0909123456

2.5 Thực hiện thao tác bằng câu lệnh SQL

3 Thao tác với SQLite3

3.1 Lấy thông tin database

```
import sqlite3

try:
    sqlite_Connection = sqlite3.connect('PythonDB.db')
    conn = sqlite_Connection.cursor()
    print("\nDatabase created and connected to SQLite.")
    sqlite_select_Query = "select sqlite_version();"
    conn.execute(sqlite_select_Query)
    record = conn.fetchall()
    print("\nSQLite Database Version is: ", record)
    conn.close()
except sqlite3.Error as error:
    print("\nError while connecting to sqlite", error)
finally:
    if (sqlite_Connection):
        sqlite_Connection.close()
        print("\nThe SQLite connection is closed.")
```

3.2 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), TenHH
char(40), MoTa char(55),DonGia decimal(10,2),SKU char(15)
NULL);")
    print("\nĐã tạo bảng HangHoa.")
    conn.commit()


sqllite_conn = sql_connection()
sql_table(sqllite_conn)
if (sqllite_conn):
    sqllite_conn.close()
    print("\nĐóng kết nối.")
```

3.3 Thực hiện thêm data cho bảng HocVien

4 Quản lý Hàng Hóa

4.1 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),
            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()
```

4.2 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()
```

4.3 Cập nhật dữ liệu

```
import sqlite3
from sqlite3 import Error
import dbcommon

def update_task(conn, task):
    """
    update priority, begin_date, and end date of a task
    :param conn:
    :param task:
    :return: project id
    """
    sql = ''' UPDATE tasks
              SET priority = ? ,
                  begin_date = ? ,
                  end_date = ?
              WHERE id = ?'''
    cur = conn.cursor()
    cur.execute(sql, task)
    conn.commit()


def main():
    database = r"C:\sqlite\db\pythonsqlite.db"

    # create a database connection
    conn = dbcommon.create_connection(database)
    with conn:
        update_task(conn, (2, '2015-01-04', '2015-01-06',
2))


if __name__ == '__main__':
    main()
```

4.4 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()
```

4.5 Xóa data

```python
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);
```

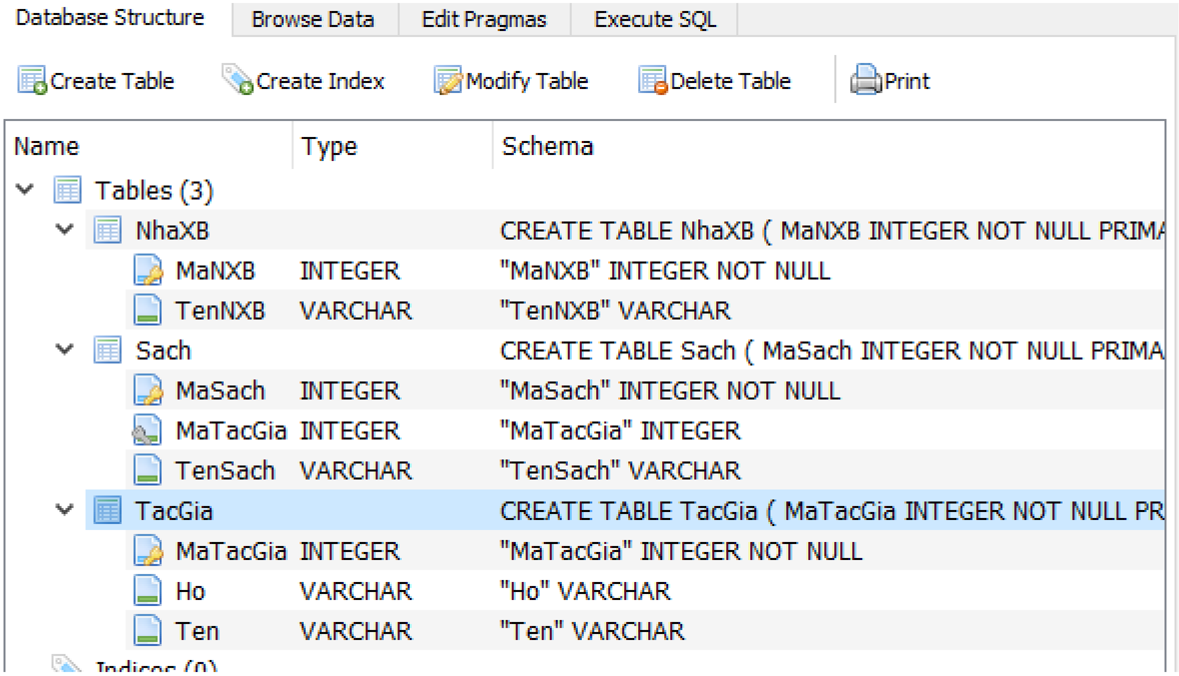
```
if __name__ == '__main__':
    main()
```

5 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
);
```



Name	Type	Schema
Tables (3)		
NhaXB		CREATE TABLE NhaXB (MaNXB INTEGER NOT NULL PRIMA
MaNXB	INTEGER	"MaNXB" INTEGER NOT NULL
TenNXB	VARCHAR	"TenNXB" VARCHAR
Sach		CREATE TABLE Sach (MaSach INTEGER NOT NULL PRIMA
MaSach	INTEGER	"MaSach" INTEGER NOT NULL
MaTacGia	INTEGER	"MaTacGia" INTEGER
TenSach	VARCHAR	"TenSach" VARCHAR
TacGia		CREATE TABLE TacGia (MaTacGia INTEGER NOT NULL PR
MaTacGia	INTEGER	"MaTacGia" INTEGER NOT NULL
Ho	VARCHAR	"Ho" VARCHAR
Ten	VARCHAR	"Ten" VARCHAR
Indices (0)		