**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM  
KHOA CÔNG NGHỆ THÔNG TIN**

**-----\*\*\*-----**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN “LẬP TRÌNH WINDOWS”**

***Đề tài:***

***XÂY DỰNG CHƯƠNG TRÌNH TRA CỨU BÀI HÁT***

***GVHD: ThS. Nguyễn Hạnh Phúc***

***Sinh viên thực hiện: Lưu Văn Quyết – Mã sv: 68037***

***Tô Hương Quỳnh – Mã sv: 70099***

***Hải Phòng, tháng 04 năm 2019***

**TRƯỜNG ĐẠI HỌC HÀNG HẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KHOA HỌC MÁY TÍNH**

**-----\*\*\*-----**

**BÀI TẬP LỚN**

**HỌC PHẦN: LẬP TRÌNH WINDOWS**

**Mã đề tài: 12**

1. **Tên đề tài**

***Xây dựng chương trình tra cứu bài hát.***

1. **Mục đích**

Phân mềm quản lý bài hát giúp người dùng dễ dàng sử dụng. Với phần mềm này, bạn có thể nhập bất cứ bộ sưu tập âm nhạc nào vào thiết bị lưu trữ mà bạn có.

Bạn có thể thêm hoặc chỉnh sửa bất kỳ loại thông tin nào liên quan đến album trong bộ sưu tập.

Sau khi lưu album, bạn có thể duyệt, lọc album theo tất cả các tiêu chí và tìm kiếm các bài hát có trong album mà bạn quan tâm.

Chương trình có thể hiển thị và in bất kỳ dữ liệu nào từ bộ sưu tập nhạc theo những cách khác nhau, vì vậy bạn có thể chọn dữ liệu mà bạn cần.

1. **Công việc cần thực hiện**

* Xây dựng được các lớp bài hát, tác giả, thể loại, album, ca sĩ.
* Xây dựng được cơ sở dữ liệu cho dự án gồm các bảng: BaiHat, CaSi, Album, TheLoai, HangSanXuat, CaSi\_BaiHat.
* Xây dựng giao diện của chương trình cho phép người dùng có thể thêm, sửa, xóa, xem thông tin chi tiết của bài hát qua giao diện.
* Xây dựng chương trình gồm các chức năng cho phép tra cứu bài hát theo tên bài hát, tên thể loại, tên tác giả, tên hãng sản xuất.
* Xây dựng các chức năng thêm, sửa, xóa thông tin có trên cơ sở dữ liệu.
* Xây dựng chức năng phân loại bài hát theo thể loại, tác giả, album, hãng sản xuất.
* Làm báo cáo bài tập lớn
* Bảo vệ bài tập lớn

1. **Yêu cầu**

* Kết quả làm bài tập lớn: Báo cáo bài tập lớn
* Báo cáo bài tập lớn phải được trình bày theo mẫu quy định (kèm theo), báo cáo có thể kết xuất thành tệp định dạng PDF và nộp qua email (không bắt buộc phải in ấn)
* Hạn nộp báo cáo bài tập lớn:

1. **Tài liệu tham khảo**

* *Bài giảng Lập trình Windows của Khoa công nghệ thông tin Trường đại học Hàng hải Việt Nam.*
* *Giáo trình của Howkteam.com*
* *Tài liệu của Linkerpt.com*

***Hải Phòng, tháng 04 năm 2019***

NGƯỜI HƯỚNG DẪN

**MỤC LỤC**

[GIỚI THIỆU 1](#_Toc6995315)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 2](#_Toc6995316)

[1.1. Ngôn ngữ C sharp 2](#_Toc6995317)

[1.1.1. Khái niệm: 2](#_Toc6995318)

[1.1.2. Các đặc trưng của ngôn ngữ C sharp 2](#_Toc6995319)

[1.1.3. Lớp và đối tượng 3](#_Toc6995320)

[1.2. Ngôn ngữ truy vấn có cấu trúc (SQL- Structured Query Language) 9](#_Toc6995321)

[1.2.1. Ngôn ngữ định nghĩa dữ liệu 9](#_Toc6995322)

[1.2.2. Ngôn ngữ thao tác dữ liệu 10](#_Toc6995323)

[CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH 12](#_Toc6995324)

[2.1. Xây dựng bảng cơ sở dữ liệu 12](#_Toc6995325)

[1.1.1. Bảng thể loại 13](#_Toc6995326)

[1.1.2. Bảng bài hát 13](#_Toc6995327)

[1.1.3. Bảng Album 13](#_Toc6995328)

[1.1.4. Bảng hãng sản xuất 14](#_Toc6995329)

[1.1.5. Bảng ca sỹ 14](#_Toc6995330)

[1.1.6. Bảng tác giả 14](#_Toc6995331)

[1.1.7. Bảng ca sỹ - bài hát 14](#_Toc6995332)

[1.1.8. Các ràng buộc toàn vẹn 15](#_Toc6995333)

[2.2. Xây dựng giao diện 17](#_Toc6995334)

[2.3. Xây dựng các lớp đối tượng. 19](#_Toc6995335)

[2.3.1. Lớp bài hát 19](#_Toc6995336)

[2.3.2. Lớp thể loại 22](#_Toc6995337)

[2.3.3. Lớp Album 23](#_Toc6995338)

[2.3.4. Lớp ca sĩ 24](#_Toc6995339)

[2.3.5. Lớp tác giả 25](#_Toc6995340)

[2.3.6. Lớp hãng sản xuất 26](#_Toc6995341)

[2.4. Xây dựng các câu truy vấn trong SQL 27](#_Toc6995342)

[2.4.1. Lớp kết nối 27](#_Toc6995343)

[2.4.2. Lớp bài hát 28](#_Toc6995344)

[2.4.3. Lớp album 29](#_Toc6995345)

[2.4.4. Lớp thể loại 29](#_Toc6995346)

[2.4.5. Lớp ca sĩ 30](#_Toc6995347)

[2.4.6. Lớp tác giả 31](#_Toc6995348)

[2.4.7. Lớp ca sĩ - tác giả 31](#_Toc6995349)

[2.4.8. Lớp hãng sản xuất 32](#_Toc6995350)

[KẾT QUẢ CHẠY CHƯƠNG TRÌNH 33](#_Toc6995351)

[KẾT LUẬN 38](#_Toc6995352)

[TÀI LIỆU THAM KHẢO 39](#_Toc6995353)

MỤC LỤC HÌNH ẢNH

[Hình 1- Toàn bộ cơ sở dữ liệu của bài toán 12](#_Toc6837380)

[Hình 2- Mối liên kết rằng buộc bảng THELOAI và bảng BAIHAT 15](#_Toc6837381)

[Hình 3-Mối liên kết rằng buộc bảng ALBUM và bảng BAIHAT 15](#_Toc6837382)

[Hình 4-Mối liên kết rằng buộc bảng HANGSANXUAT và bảng BAIHAT 16](#_Toc6837383)

[Hình 5-Mối liên kết rằng buộc bảng TACGIA và bảng BAIHAT 16](#_Toc6837384)

[Hình 6-Mối liên kết rằng buộc bảng CASI\_BAIHAT và bảng BAIHAT 17](#_Toc6837385)

[Hình 7-Mối liên kết rằng buộc bảng CASI\_BAIHAT và bảng CASI 17](#_Toc6837386)

[Hình 16-Kết quả Load page home 33](#_Toc6837387)

[Hình 17- Kết quả load page ALBUM 33](#_Toc6837388)

[Hình 18- Kết quả load page THỂ LOẠI 34](#_Toc6837389)

[Hình 19- Kết quả load page CA SĨ 34](#_Toc6837390)

[Hình 20- Kết quả load page TÁC GIẢ 35](#_Toc6837391)

[Hình 21- Kết quả load page TRA CỨU – TÌM KIẾM 35](#_Toc6837392)

[Hình 22- Kết quả load page BÀI HÁT 36](#_Toc6837393)

[Hình 23- Kết quả load Form thêm bài hát mới 36](#_Toc6837394)

[Hình 24-Kết quả load form cập nhật bài hát 37](#_Toc6837395)

[Hình 25- Kết quả load Form chi tiết bài hát 37](#_Toc6837396)

# GIỚI THIỆU

***Bài toán:***

Xây dựng chương trình tra cứu bài hát cho phép người dùng thao tác các sau:

* + Tra cứu bài hát theo tên bài hát, tên album, tên thể loại, tên ca sĩ, tên tác giả, tên hãng sản xuất, lời bài hát.
  + Thực hiện các chức năng thêm, sửa, xóa các thông tin của bài hát.
  + Lưu trữ bài hát (ít nhất 200 bài).
  + Cho phép người dùng nghe nhạc trên giao diện của chương trình.

***Yêu cầu:***

* + Xây dựng được giao diện phù hợp với bài toán đặt ra.
  + Thiết kế được cơ sở dữ liệu lưu trữ các thông tin về bài hát.
  + Thực hiện được các yêu cầu của bài toán đặt ra trong việc thêm, sửa, xóa, tra cứu thông tin của bài hát.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## Ngôn ngữ C sharp

### Khái niệm:

Ngôn ngữ c sharp hay còn được biết đến là C# là ngôn ngữ lập trình hiện đại, hướng đối tượng được kế thừa chủ yếu từ hai ngôn ngữ C++ và Java.

### Các đặc trưng của ngôn ngữ C sharp

#### C# là ngôn ngữ đơn giản.

Nếu chúng ta từng học C, C++ hay Java thì chúng ta có thể thấy về giao diện, cú pháp, biểu thức, toán tử, và các chức năng khác khá quen thuộc. Bởi vì, C# là được phát triển chủ yếu dựa trên nền tảng của 2 ngôn ngữ chủ yếu là C++ và Java. Một vài sự cải tiển trong C# đó là loại bỏ các dư thừa và thêm vào những cú pháp thay đổi.

#### C# là ngôn ngữ hiện đại

Bởi vì C# chứa các tính năng của một ngôn ngữ hiện đại như xử lý ngoại lệ, những kiểu dữ liệu mở rộng, bảo mật mã nguồn…

#### C# là ngôn ngữ lập trình thuần hướng đối tượng

C# có đầy đủ các đặc tính của một ngôn ngữ lập trình hướng đối tượng bao gồm tính trừu tượng(abstraction), tính đóng gói(encapsualation), tính đa hình (polymorphism) và tính kế thừa (inheritance).

#### Ưu điểm của C#

* C# khá gần gũi với các ngôn ngữ truyền thống là C và C++ nên việc tiếp cận và học cũng dễ dàng hơn.
* C# được kế thừa những ưu điểm của 2 ngôn ngữ C++ và Java.
* C# có .Netframework hỗ trợ mạnh mẽ cho người lập trình viên.
* C# có thể được biên dịch trên nhiều nền tảng máy tính khác nhau.

### Lớp và đối tượng

#### Lớp

Lớp là tập hợp các đối tượng có các thuộc tính dữ liệu cùng kiểu và các hành vi giống nhau.

Cũng giống như trong C++ để khai báo một lớp ta cần sử dụng từ khóa class theo sau là tên của class. Bên trong lớp bao gồm các biến và các phương thức được tạo.

Vd: Xây dựng một lớp sinh viên với các thuộc tính: tên sinh viên, mã sinh viên, điểm toán, lý, hóa và các phương thức: khởi tạo mặc định, hàm tính điểm trung bình.

public class Sinhvien

{

private string hoten;

private string masv;

private float dtoan;

private float dhoa;

private float dly;

public Sinhvien();

public float GetDTB();

}

#### Đối tượng

Đối tượng là một thực thể (entity) bao gồm một tập hợp các thuộc tính (properties): thuộc tính dữ liệu (data member) và các hành động (behavior). Thuộc tính dữ liệu nói nên trạng thái của đối tượng, hành động nói đến khả năng thực hiện công việc của đối tượng.

Trong C# tất cả các biến đều là đối tượng, các biến kiểu số, kiểu chuỗi đều đối tượng. Mỗi đối tượng đều có các thuộc tính và các phương thức (hàm) để tác động lên các biến thành viên (các thuộc tính).

Vd: Khai báo một đối tượng có lớp có sẵn.

string str1 = "";

int a;

Vd2: Khai báo một đối tượng một lớp tự định nghĩa.

public class Sinhvien

{

private string hoten;

private string masv;

private float dtoan;

private float dhoa;

private float dly;

public Sinhvien();

public float GetDTB();

}

public class main

{

Sinhvien sv = new Sinhvien();

}

#### Các phạm vi truy cập

|  |  |
| --- | --- |
| Phạm vi truy cập | Ý nghĩa |
| public | Các thành phần được gắn thuộc tính có thể được truy xuất mọi nơi ( thường gắn cho các phương thức). |
| protected | Được sử dụng nội bộ trong lớp và lớp con ( thường sử dụng khi có kế thừa). |
| private | Từ khóa private được thể hiện sự riêng tư và chỉ có thể truy cập ở trong lớp ( thường dùng cho các biến của lớp). |
| static | Thành phần được gắn static khi truy xuất sẽ không phải khởi tạo đối tượng ở trong lớp ( thường được sử dụng làm biến đếm). |
| internal | Các thành phần được gắn internal chỉ có thể được truy xuất trong cùng một project. |
| protected internal | Cũng giống như internal nhưng nó còn được truy xuất trong lớp con ( lớp dẫn xuất). |

#### Từ khóa get và set

Từ khóa get và set được sự dụng để thay thế cho phương thức truy vấn và phương thức cập nhật, khi được nó sẽ tạo ra một kiểu cấu trúc mới có tên là property.

Cú pháp:

<Kiểu dữ liệu> <Tên property>

{

Get { return <tên thuộc tính>}

Set { <ten thuộc tính> = value;}

}

Trong đó:

* <Kiểu dữ liệu>: Tên kiểu dữ liệu của property, thường là kiểu dữ liệu của các thuộc tính private của lớp.
* <Tên property>: tên do người dùng đặt.
* Từ khóa get: tương đương với phương thức truy vấn.
* Từ khóa set: tương đương với phương thức cập nhật.
* Từ khóa value: giá trị mà người dùng gắn.

Ví dụ: Điểm toán trong lớp sinh viên cần phải có tính đóng gói. Khi đó ta sẽ sử dụng cấu trúc property cho thuộc tính này như sau:

Private float Diemtoan;

Public float DiemToan

{

Get { return this.Diemtoan;}

Set { Diemtoan = value;}

}

#### Tính kế thừa

Trong lập trình kế thừa được hiểu là một lớp mới tạo ra sẽ được thừa hưởng những thuộc tính và phương thức mà lớp cha để lại (lớp cơ sở). Những thuộc tính và phương thức này được sử dụng bình thường trong lớp mới.

Cú pháp:

**Class** <Tên lớp con> : <Tên lớp cha>

{

}

Ví dụ: Xây dựng lớp tứ giác được kế thừa từ lớp tam giác

class Tamgiac

{

protected int a;

protected int b;

protected int c;

public Tamgiac(int a1,int b1,int c1)

{

a = a1;

b = b1;

c = c1;

}

public int chuvi()

{

return (a + b + c);

}

}

class Hinhtugiac : Tamgiac

{

private int d;

public Hinhtugiac(int a1, int b1, int c1,int d1): base(a1, b1, c1)

{

d = d1;

}

}

Từ khóa **base** được sự dụng để gọi các constructor của lớp cha.

#### Tính đa hình

Đa hình là một kĩ thuật cho phép một lời gọi hàm có thể gọi đến nhiều hàm thuộc nhiều lớp khác nhau.

Để thể hiện được tính đa hình phải cần các điều kiện sau:

* Các lớp phải có quan hệ cha con (tính kế thừa).
* Phương thức đa hình phải được ghi đè (override) ở các lớp con.

*Từ khóa* ***virtual*** *và* ***override***

* **Virtual** là từ được dùng để khai báo một phương thức ảo (phương thức sẽ được ghi đè).
* **Override** là từ khóa dùng để chỉ phương thức ghi đè lên phương thức của lớp cha.

Ví dụ: Xây dựng lớp tứ giác kế thừa từ lớp tam giác có sử dụng tính đa hình.

class Tamgiac

{

protected int a;

protected int b;

protected int c;

public Tamgiac(int a1, int b1, int c1)

{

a = a1;

b = b1;

c = c1;

}

public virtual int chuvi()

{

return (a + b + c);

}

}

class Hinhtugiac : Tamgiac

{

private int d;

public Hinhtugiac(int a1, int b1, int c1, int d1)

: base(a1, b1, c1)

{

d = d1;

}

public override int chuvi()

{

return base.chuvi() + d;

}

}

class Program

{

static void Main(string[] args)

{

Tamgiac t1 = new Tamgiac(2, 4, 5);

Console.WriteLine("Chu vi cua tam giac là:" + t1.chuvi());

Tamgiac t2 = new Hinhtugiac(4, 6, 7, 8);

Console.WriteLine("Chu vi cua hinh tu giac là:" + t2.chuvi());

Console.ReadLine();

}

}

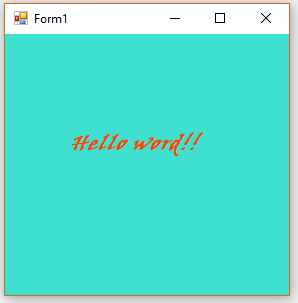
Kết quả:



#### Windows Forms

* Windows Forms là cách cơ bản để cung cấp các thành phần giao diện (GUI compenents) cho môi trường .NET Framework.
* Windows Forms được xây dựng trên thư viện Windows API.
* Windows Forms cơ bản bao gồm:
* Một Form là khung dùng hiển thị thông tin đến người dùng.
* Một Control được đặt trong Form và được lập trình để đáp ứng sự kiện.

Ví dụ: Xây dựng chương trình hiện thị chữ “hello word!” ra form.



private void Form1\_Load(object sender, EventArgs e)

{

label1.Text = "Hello word!!";

}

## Ngôn ngữ truy vấn có cấu trúc (SQL- Structured Query Language)

SQL gồm 2 thành phần

* Ngôn ngữ đinh nghĩa dữ liệu
* Ngôn ngữ thao tác dữ liệu

### Ngôn ngữ định nghĩa dữ liệu

#### Cách tạo một database

**Cú pháp**: Create database <tên database>

Ví dụ:

create database QuanLiNhac

#### Cách tạo bảng dữ liệu

**Cú pháp:**

create table <tên bảng>

(

<Các thuộc tính bên trong>

)

Ví dụ: Xây dựng bảng ALBUM với các thuộc tính mã album, tên album.

create table ALBUM

(

MaAlbum varchar(10) not null,

TenAlbum nvarchar(100) not null,

NamPhatHanh char(4) not null

constraint PK\_ALBUM primary key (MaAlbum)

)

### Ngôn ngữ thao tác dữ liệu

#### Câu lệnh thêm bản ghi

**Cú pháp:**

INSERT INTO <TÊN BẢNG> VALUES (<DS GIÁ TRỊ>)

Ví dụ:

INSERT INTO ALBUM VALUES('AL004',N'Ngày trở về')

#### Câu lệnh cập nhật dữ liệu

**Cú pháp:**

UPDATE <TÊN BẢNG>

SET <Tên cột 1> = <giá trị 1>, <tên cột 2> = <giá trị 2>,..

Where <điều kiện>

Ví dụ:

UPDATE PhongBan

SET TenPB = N'Phòng Kế hoạch – Tài chính', DiaDiem = N'P302 - Nhà A6'

WHERE MaPB = 'PTV'

#### Câu lệnh xóa dữ liệu

**Cú pháp:**

DELETE FROM <tên bảng> WHERE <điều kiện>

Ví dụ:

DELETE FROM NhanVien WHERE ChuyenMon = N'Kỹ sư tin học'

#### Mệnh đề select

**Cú pháp:**

SELECT \*|<danh sách các cột> FROM <tên bảng> Where <điều kiện>

Ví dụ: Liệt kê các học sinh của lớp 10A1 có điểm trên 8 dưới 6

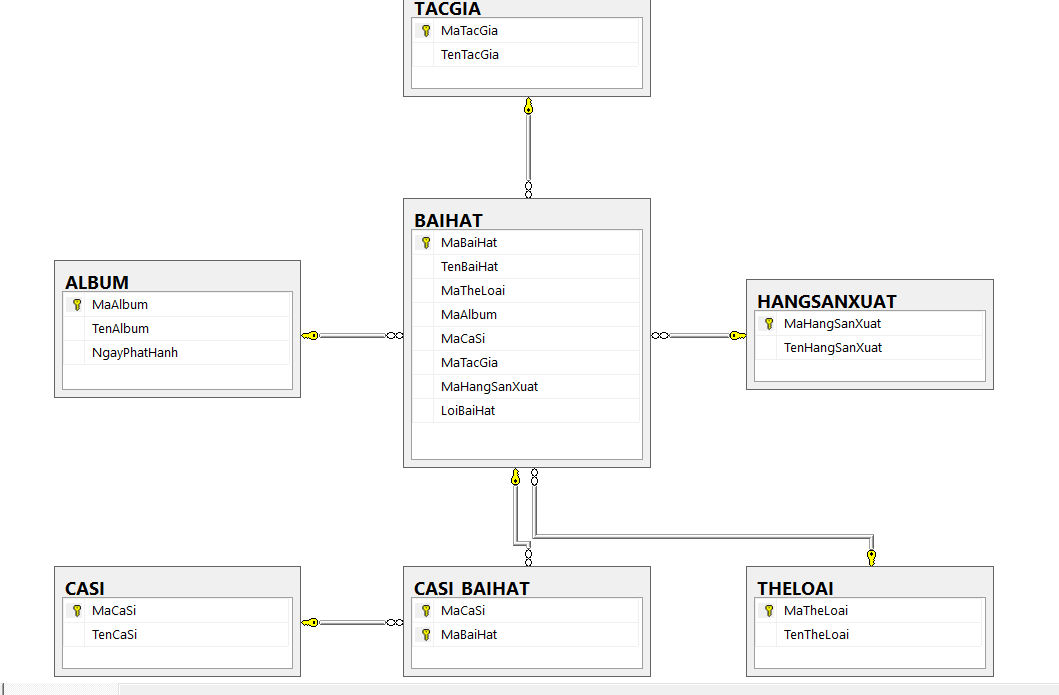
SELECT MaHS, TenHS, Diem FROM HocSinh

WHERE ((Diem >= 8) OR (Diem < 6)) AND (Lop = '10A1')

# CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH

## Xây dựng bảng cơ sở dữ liệu

Cơ sở dữ liệu được xây dựng trên phần mềm quản trị cơ sở dữ liệu SQL Server.



Hình 1- Toàn bộ cơ sở dữ liệu của bài toán

Cơ sở dữ liệu được chia làm 7 bảng:

* Bảng thể loại: THELOAI.
* Bảng hãng sản xuất: HANGSANXUAT.
* Bảng ca sĩ: CASI.
* Bảng album: ALBUM.
* Bảng tác giả: TACGIA.
* Bảng ca sĩ – bài hát: CASI\_BAIHAT.
* Bảng bài hát: BAIHAT.

### Bảng thể loại

Chứa các thông tin về thể loại gồm mã thể loại, tên thể loại.

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liều** | **Ghi chú** |
| 1 | MaTheLoai | varchar(10) | Khóa chính |
| 2 | TenTheLoai | Nvarchar(50) | Tên thể loại |

### Bảng bài hát

Chứa thông tin mã bài hát, tên bài hát, mã thể loại, mã album, mã tác giả, mã ca sĩ, mã hãng sản xuất, lời bài hát.

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liều** | **Ghi chú** |
| 1 | MaBaiHat | varchar(10) | Khóa chính |
| 2 | TenBaiHat | Nvarchar(100) | Tên bài hát |
| 3 | MaTheLoai | varchar(10) | Khóa phụ |
| 4 | MaAlbum | varchar(10) | Khóa phụ |
| 5 | MaTacGia | varchar(10) | Khóa phụ |
| 6 | MaCaSy | varchar(10) | Khóa phụ |
| 7 | MaHangSanXuat | varchar(10) | Khóa phụ |
| 8 | LoiBaiHat | Nvarchar | Lời bài hát |

### Bảng Album

Chứa thông tin mã album, tên album, ngày phát hành.

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liều** | **Ghi chú** |
| 1 | MaAlbum | varchar(10) | Khóa chính |
| 2 | TenAlbum | Nvarchar(50) | Tên Album |
| 3 | NamPhatHanh | Char(4) | Ngày phát hành |

### Bảng hãng sản xuất

Chứa các thông tin như mã hãng sản xuất, tên hãng sản xuất.

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liều** | **Ghi chú** |
| 1 | MaHangSanXuat | varchar(10) | Khóa chính |
| 2 | TenHangSX | Nvarchar(50) | Tên hãng sản xuất |

### Bảng ca sỹ

Chứa các thông tin như mã ca sĩ, tên ca sĩ.

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liều** | **Ghi chú** |
| 1 | MaCaSi | varchar(10) | Khóa chính |
| 2 | TenCaSi | Nvarchar(50) | Tên ca sĩ |

### Bảng tác giả

Chứa các thông tin như mã tác giả, tên tác giả.

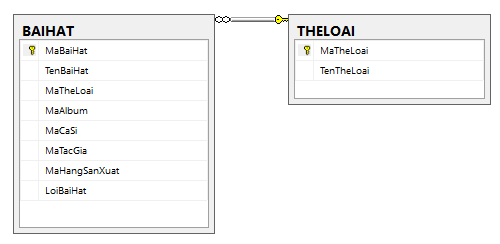
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liều** | **Ghi chú** |
| 1 | MaTacGia | varchar(10) | Khóa chính |
| 2 | TenTacGia | Nvarchar(50) | Tên tác giả |

### Bảng ca sỹ - bài hát

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liều** | **Ghi chú** |
| 1 | MaBaiHat | varchar(10) | Khóa chính |
| 2 | MaCaSi | Nvarchar(50) | Khóa chính |

### Các ràng buộc toàn vẹn

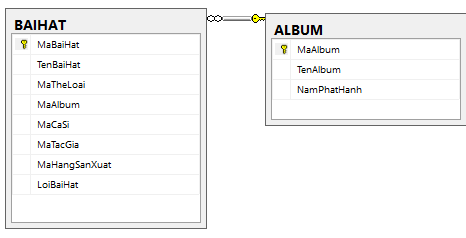
*Bảng THELOAI và bảng BAIHAT*



Hình 2- Mối liên kết rằng buộc bảng THELOAI và bảng BAIHAT

Bảng THELOAI liên kết với bảng BAIHAT theo mối quan hệ 1 nhiều, bởi vì một thể loại bao gồm nhiều bài hát. Hai bảng này liên kết với nhau thông qua MaTheLoai.

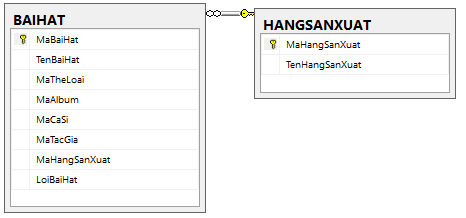
*Bảng ALBUM và bảng BAIHAT*



Hình 3-Mối liên kết rằng buộc bảng ALBUM và bảng BAIHAT

Bảng ALBUM liên kết với bảng BAIHAT theo mối quan hệ 1 nhiều, bởi vì Album có thể bao gồm nhiều bài hát. Hai bảng này liên kết với nhau thông qua MaAlbum.

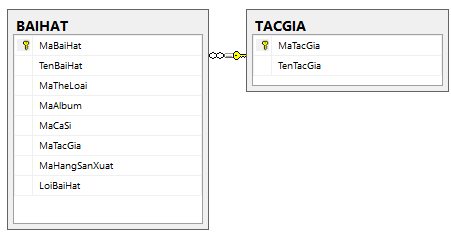
*Bảng HANGSANXUAT và bảng BAIHAT*



Hình 4-Mối liên kết rằng buộc bảng HANGSANXUAT và bảng BAIHAT

Bảng HANGSANXUAT liên kết với bảng BAIHAT theo mối quan hệ 1 nhiều, bởi vì một hãng sản xuất có thể sản xuất ra nhiều bài hát. Hai bảng này liên kết với nhau thông qua MaHangSanXuat.

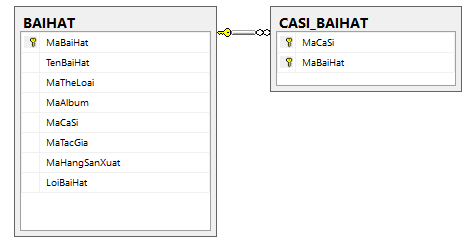
Bảng TACGIA và bảng BAIHAT



Hình 5-Mối liên kết rằng buộc bảng TACGIA và bảng BAIHAT

Bảng TACGIA liên kết với bảng BAIHAT theo mối quan hệ 1n nhiều, bởi vì một tác giả có thể sáng tác nhiều bài hát. Hai bảng này liên kết với nhau thông qua MaTacGia.

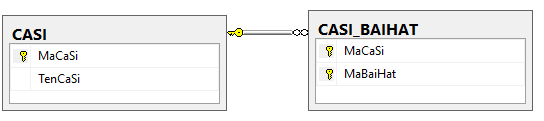
*Bảng BAIHAT và bảng CASI\_BAIHAT*



Hình 6-Mối liên kết rằng buộc bảng CASI\_BAIHAT và bảng BAIHAT

Bảng BAIHAT liên kết với bảng CASI\_BAIHAT theo mối quan hệ 1 nhiều, bởi vì một bài hát có thể do nhiều ca sĩ thể hiện. Hai bảng này liên kết với nhau thông qua MaBaiHat.

*Bảng CASI\_BAIHAT và bảng CASI*



Hình 7-Mối liên kết rằng buộc bảng CASI\_BAIHAT và bảng CASI

Bảng CASI liên kết với bảng CASI\_BAIHAT theo mối quan hệ 1 nhiều, bởi vì một ca sĩ có thể thể hiện nhiều bài hát. Hai bảng này liên kết với nhau thông qua MaCaSi.

## Xây dựng giao diện

Giao diên chương trình chia làm 14 form.

Form chính gồm 7 page:

* ***Page 1:*** Hiện thị bảng bài hát bao gồm các cột: STT, Tên bài hát, tên thể loại, tên album, năm phát hành, tên ca sĩ, tên tác giả, tên hãng sản xuất, lời bài hát.
* ***Page 2:*** Gồm 2 bảng hiển thị: bảng album (STT, Tên album, Năm phát hành), bảng bài hát (STT, Tên bài hát, các ca sĩ thể hiện, tên thể loại, lời bài hát). Có 3 phím chức năng: Thêm album mới, chỉnh sửa thông tin album, xóa album.
* ***Page 3:*** Gồm 2 bảng hiển thị: bảng thể loại (STT, Tên thể loại), bảng bài hát (STT, Tên bài hát, tên album, các ca sĩ thể hiện, lời bài hát). Có 3 nút chức năng: Thêm thể loại mới, chỉnh sửa thông tin thể loại, xóa thể loại.
* ***Page 4:*** Gồm 2 bảng hiển thị: bảng ca sĩ (tên ca sĩ), bảng bài hát (STT, Tên bài hát). Có 3 nút chức năng: Thêm bài ca sĩ mới, chỉnh sửa thông tin ca sĩ, xóa ca sĩ.
* ***Page 5:*** Gồm 2 bảng hiển thị: bảng tác giả (tên tác giả), bảng bài hát (STT, Tên bài hát). Có 3 nút chức năng: Thêm tác giả mới, chỉnh sửa thông tin tác giả, xóa tác giả.
* ***Page 6:*** Hiện thị bảng kết quả tra cứu bài hát theo tên bài hát, tên thể loại, tên album, tên ca sĩ thể hiện, tên tác giả, tên hãng sản xuất gồm các thông tin hiện thị: STT, tên bài hát, tên thể loại, tên album, tên ca sĩ, tên tác giả, tên hãng sản xuất, lời bài hát.
* ***Page 7:*** Gồm bảng hiện thị: bảng bài hát (STT, tên bài hát, lời bài hát) và phần tra cứu bài hát theo cả 5 tiêu chí: tên thể loại, tên album, tên ca sĩ, tên tác giả, tên hãng sản xuất. Có 3 nút chức năng: Thêm bài hát mới, chỉnh sửa thông tin bài hát, xóa bài hát.

## Xây dựng các lớp đối tượng.

Chương trình bao gồm 6 lớp đối tượng chính bao gồm: lớp bài hát, lớp album, lớp thể loại, lớp ca sĩ, lớp tác giả, lớp hãng sản xuất.

### Lớp bài hát

Tên lớp: tblBaihat

Các thuộc tính:

* Mã bài hát: mabaihat
* Tên bài hát: tenbaihat
* Mã thể loại: matheloai
* Mã album: maalbum
* Mã ca sĩ: macasi
* Mã tác giả: matacgia
* Mã hãng sản xuất: hangsanxuat

Các phương thức:

* Các hàm khởi tạo: khởi tạo mặc định, khởi tạo có tham số
* Hàm lấy mã bài hát: MABAIHAT
* Hàm lấy tên bài hat: TENBAIHAT
* Hàm lấy mã thể loại: MATHELOAI
* Hàm lấy mã Album: MAALBUM
* Hàm lấy mã ca sĩ: MACASI
* Hàm lấy mã tác giả: MATACGIA
* Hàm lấy mã hãng sản xuất: MAHANGSANXUAT
* Các hàm lấy bảng dữ liệu trong cơ sở dữ liệu

Thực thi:

class tblBaihat

{

Baihat\_Data objBaiHat = new Baihat\_Data();

#region khai báo đối tượng BaiHat và kiểm tra hợp lệ

private string mabaihat;

private string tenbaihat;

private string matheloai;

private string maalbum;

private string macasi;

private string matacgia;

private string mahangsanxuat;

private string loibaihat;

//Khởi tạo mặc định

public tblBaihat() { }

//Khởi tạo có tham số

public tblBaihat(string ma, string ten, string matl, string maal, string macs, string matg, string masx, string loi){…}

//Khởi tạo có tham số

public tblBaihat(string ma){…}

//Lấy giá trị mã bài hát và qui định phạm vi của biến mabaihat

public string MABAIHAT{…}

//Lấy giá trị tên bài hát và qui định phạm vi của biến tenbaihat

public string TENBAIHAT{…}

//Lấy giá trị lời bài hát và qui định phạm vi của biến loibaihat

public string LOIBAIHAT{…}

#endregion

#region các phương thức xử lý

//Kiểm tra câu lệnh thêm bài hát mới

public int themBaiHat(){…}

//Kiểm tra câu lệnh xóa bài hát

public int xoaBaiHat(){…}

//Kiểm tra câu lệnh cập nhật bài hát

public int capnhatBaiHat(){…}

#endregion

#region các hàm lấy dử liệu bài hát (table)

//Hàm lấy bảng dữ liệu bài hát

public DataTable getBaiHat(){…}

//Hàm lấy bảng dữ liệu bài hát ở home

public DataTable getBaiHat\_home(){…}

//Hàm lấy bảng dữ liệu bài hát qua mã album

public DataTable getBaiHat\_by\_album(string maalbum){…}

//Hàm lấy bảng dữ liệu bài hát qua mã thể loại

public DataTable getBaiHat\_by\_TheLoai(string matheloai){…}

//Hàm lấy bảng dữ liệu bài hát qua mã bài hát

public DataTable getBaiHat\_by\_mabaihat(string mabaihat){…}

//Hàm lấy bảng dữ liệu bài hát qua mã thể loại,mã album, mã ca sĩ, mã tác giả, mã hãng sản xuất

public DataTable getBaiHat\_cbo(string matheloai, string maalbum, string macasi, string matacgia, string mahangsanxuat){…}

//Hàm lấy bảng dữ liệu bài hát qua mã thể loại,mã album

public DataTable getBaiHat\_cbo(string matheloai, string maalbum){…}

//Hàm lấy bảng dữ liệu bài hát qua tên bài hát

public DataTable getBaiHat\_by\_Tenbaihat(string tenbh){…}

//Hàm lấy bảng dữ liệu bài hát qua tên thể loại

public DataTable getBaiHat\_by\_Tentheloai(string tentheloai){…}

//Hàm lấy bảng dữ liệu bài hát qua tên album

public DataTable getBaiHat\_by\_Tenalbum(string tenalbum){…}

//Hàm lấy bảng dữ liệu bài hát qua tên ca sĩ

public DataTable getBaiHat\_by\_TenCasi(string tencasi){…}

//Hàm lấy bảng dữ liệu bài hát qua tên tác giả

public DataTable getBaiHat\_by\_TenTacgia(string tentacgia){…}

//Hàm lấy bảng dữ liệu bài hát qua tên hãng sản xuất

public DataTable getBaiHat\_by\_Tenhangsanxuat(string tenhangsx){…}

//Hàm lấy bảng dữ liệu bài hát qua lời bài hát

public DataTable getBaiHat\_by\_Loibaihat(string loibaihat){…}

#endregion

}

### Lớp thể loại

Tên lớp: tblTheloai

Các thuộc tính:

* Mã thể loại: matheloai
* Tên thể loại: tentheloai

Các phương thức:

* Các hàm khởi tạo: khởi tạo mặc định, khởi tạo có tham số.
* Lấy mã thể loại: MATHELOAI
* Lấy tên thể loại: TENTHELOAI

Thực thi:

class tblTheloai

{

Theloai\_Data objTheLoai = new Theloai\_Data();

#region khai báo đối tượng và kiểm tra hợp lệ

private string matheloai;

private string tentheloai;

public tblTheloai() { }

public tblTheloai(string ma, string ten){…}

public tblTheloai(string ma){…}

public string MATHELOAI{…}

public string TENTHELOAI{…}

#engion

#region các hàm lấy dử liệu bảng

public DataTable getTheLoai(){…}

public DataTable getTheLoai\_by\_ma(string matheloai){…}

public DataTable getTheLoai\_by\_mabaihat(string mabaihat){…}

#endregion

#region các phương thức xử lý

public int themTheLoai(){…}

public int xoaTheLoai(){…}

public int capnhatTheLoai(){…}

#endregion

}

### Lớp Album

Tên lớp: tbl Album

Các thuộc tình:

* Mã album: maalbum
* Tên album: tenalbum
* Năm phát hành: namphathanh

Các phương thức:

* Các hàm khởi tạo: khởi tạo mặc định, khởi tạo có tham số.
* Hàm lấy mã album: MAALBUM.
* Hàm lấy tên album: TENALBUM.
* Hàm lấy năm phát hành: NAMPHATHANH
* Hàm lấy dữ liệu bài hát qua mã album.

Thực thi:

class tblAlbum

{

Album\_Data objAlbum = new Album\_Data();

#region khai báo đối tượng, kiểm tra hợp lệ

private string maalbum;

private string tenalbum;

private string Namphathanh;

public tblAlbum(string ma, string ten, string namphathanh){…}

public tblAlbum() { }

public tblAlbum(string ma){…}

public string MAALBUM{…}

public string TENALBUM{…}

public string NAMPHATHANH{…}

#region các hàm lấy dử liệu bảng album và các phương thức xử lý

public DataTable getAlbum(){…}

public DataTable getAlbum\_by\_ma(string maalbum){…}

public int themAlbum(){…}

public int xoaAlbum(){…}

public int capnhatalbum(){…}

public int xoaBaiHat\_by\_Album(){…}

#endregion

}

### Lớp ca sĩ

Tên lớp: tblCasi

Các thuộc tính:

* Mã ca sĩ: macasi
* Tên ca sĩ: tencasi

Các phương thức:

* Các hàm khởi tạo: khởi tạo mặc định, khởi tạo có tham số.
* Hàm lấy mã ca sĩ: MACASI
* Hàm lấy tên ca sĩ: TENCASI
* Các hàm lấy dữ liệu bài hát thông qua mã ca sĩ.

Thực thi:

class tblCaSi

{

Casi\_Data objCaSi = new Casi\_Data();

#region khai báo đối tượng, kiểm tra hợp lệ

private string macasi;

private string tencasi;

public tblCaSi() { }

public tblCaSi(string ma, string ten){…}

public tblCaSi(string macasi){…}

public string MACASI{…}

public string TENCASI{…}

#endregion

#region hàm lấy dử liệu bảng và các phương thức xử lý

public DataTable getCaSi(){…}

public DataTable getCasi\_by\_macasi(string macasi){}

public int themCaSi(){…}

public int xoaCasi(){…}

public int capnhapCaSi(){…}

#endregion

### Lớp tác giả

Tên lớp: tblTacgia

Các thuộc tính:

* Mã tác giả: matacgia
* Tên tác giả: tentacgia

Các phương thức:

* Các hàm khởi tạo: khởi tạo mặc định, khởi tạo có tham số.
* Hàm lấy mã tác giả: MATACGIA
* Hàm lấy tên tác giả: TENTACGIA
* Hàm lấy dữ liệu bài hát qua mã tác giả.

Thực thi:

class tblTacgia

{

Tacgia\_Data objTacGia = new Tacgia\_Data();

#region khai báo đối tượng, kiểm tra hợp lệ

public string matacgia;

public string tentacgia;

public tblTacgia() { }

public tblTacgia(string ma){…}

public tblTacgia(string ma, string ten){…}

public string TENTACGIA{…}

#endregion

#region hàm lấy dử liệu bảng và các phương thức xử lý

public DataTable getTacGia(){…}

public int themTacGia(){…}

public int xoaTacGia(){…}

public int capnhapTacgia(){…}

#endregion

}

### Lớp hãng sản xuất

Tên lớp: tblHangsanxuat

Các thuộc tính:

* Mã hãng sản xuất: mahangsanxuat
* Tên hãng sản xuất: tenhangsanxuat

Các phương thức:

* Các hàm khởi tạo: khởi tạo mặc định, khởi tạo có tham số.
* Hàm lấy mã hãng sản xuất: MAHANGSANXUAT
* Hàm lấy tên hãng sản xuất: TENHANGSANXUAT
* Hàm lấy dữ liệu bài hát qua mã hãng sản xuất.

Thực thi:

class tblHangsanxuat

{

Hangsanxuat\_Data objNhaSanXUat = new Hangsanxuat\_Data();

#region khai báo đối tượng , kiểm tra hợp lệ

private string mahangsanxuat;

private string tenhangsanxuat;

public tblHangsanxuat() { }

public tblHangsanxuat(string ma){…}

public tblHangsanxuat(string ma, string ten){…}

public string MAHANGSANXUAT{…}

public string TENHANGSANXUAT{…}

#endregion

#region lấy bảng dử liệu và các phươnng thức xử lý

public DataTable getNhaSanXuat(){…}

public int themNhaSX(){…}

public int xoaNhaSX(){…}

public int capnhapNhaSX(){…}

#endregion

}

## Xây dựng các câu truy vấn trong SQL

Gồm có 7 lớp dữ liệu: lớp bài hát, lớp album,lớp kết nối, lớp thể loại, lớp ca sĩ, lớp tác giả, lớp hãng sản xuất, lớp ca sĩ bài hát.Tương ứng với 7 bảng trong cơ sở dữ liệu.

### ***Lớp kết nối***

Gồm phương thức kết nối với database, hàm lấy dữ liệu của bảng, hàm kiểm tra thực hiện lệnh.

Thực thi:

class DataConection

{

public SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;Initial Catalog=QuanLiNhac;Integrated Security=True");

public DataConection() { }

public DataSet getAllTable(string tableName){…}

public DataSet getBaiHat\_home(){…}

public int executeNonQuery(string sql){…}

}

### Lớp bài hát

Gồm các phương thức lấy bảng dữ liệu bài hát theo mã bài hát, mã ca sĩ, mã tác, mã thể loại, mã album, mã hãng sản xuất, tên bài hát, tên thể loại, tên album, tên ca sĩ, tên tác giả, tên hãng sản xuất, lời bài hát. Các phương thức: thêm bài hát, cập nhật thông tin bài hát theo mã bài hát, xóa bài hát.

Thực thi:

class Baihat\_Data

{

DataConection objCon = new DataConection();

#region các hàm lấy dử liệu table

public DataTable getBaiHat(){…}

public DataTable getBaiHat\_Home(){…}

public DataTable getBaiHat\_by\_Album(string maalbum){…}

public DataTable getBaiHat\_by\_mabaihat(string mabaihat){…}

public DataTable getBaiHat\_by\_TheLoai(string matheloai){…}

public DataTable getBaiHat\_by\_cbo(string matheloai, string maalbum, string macasi, string matacgia, string mahangsanxuat){…}

public DataTable getBaiHat\_by\_cbo(string matheloai, string maalbum){…}

public DataTable getBaihat\_by\_tenbaihat(string tenbh){…}

public DataTable getBaihat\_by\_tentheloai(string tentheloai){…}

public DataTable getBaihat\_by\_tenalbum(string tenalbum){…}

public DataTable getBaihat\_by\_tencasi(string tencasi){…}

public DataTable getBaihat\_by\_tentacgia(string tentacgia){…}

public DataTable getBaihat\_by\_tenhangsanxuat(string tenhangsanxuat){…}

public DataTable getBaihat\_by\_loibaihat(string loibaihat){…}

#endregion

#region các phương thức xử lý

public int themBaiHat(string mabaihat, string tenbaihat, string matheloai, string maalbum, string macasi, string matacgia, string manhasanxuat, string loibaihat){…}

public int xoaBaiHat(string mabaihat){…}

public int capnhatBaiHat(string mabaihat, string tenbaihat, string matheloai, string maalbum, string macasi, string matacgia, string mahangsanxuat, string loibaihat){…}

#endregion

}

### Lớp album

Gồm các phương thức lấy bảng album theo mã album và các phương thức thêm, sửa, xóa album.

Thực thi:

class Album\_Data

{

DataConection objCon = new DataConection();

public Album\_Data() { }

#region hàm lấy bảng album

public DataTable getAlbum(){…}

public DataTable getAlbum\_by\_ma(string maalbum){…}

#endregion

#region các phương thức xử lý

public int themAlbum(string maalbum, string tenalbum, string namphathanh){…}

public int xoaAlbum(string maalbum){…}

public int capnhatAlbum(string maalbum, string tenalbum, string namphathanh){…}

public int xoaBaiHatKhoiAlBum(string maalbum){…}

#endregion

}

### Lớp thể loại

Gồm các phương thức lấy bảng thể loại theo mã thể loại và các phương thức thêm, sửa, xóa thể loại.

Thực thi:

class Theloai\_Data

{

DataConection objCon = new DataConection();

#region các hàm lấy table, lấy hết table và lấy theo điều kiện

public DataTable getTheLoai(){…}

public DataTable getTheLoai\_by\_ma(string matheloai){…}

public DataTable getTheLoai\_by\_mabaihat(string mabaihat){…}

#endregion

#region các phương thức xử lý (câu sql) thêm xóa sửa

public int themTheLoai(string matheloai, string tentheloai){…}

public int xoaTheLoai(string matheloai){…}

public int capnhatTheLoai(string matheloai, string tentheloai){…}

#endregion

}

### Lớp ca sĩ

Gồm các phương thức lấy bảng dữ liệu ca sĩ theo mã ca sĩ và các phương thức thêm, sửa, xóa ca sĩ.

Thực thi:

class Casi\_Data

{

DataConection objCon = new DataConection();

#region hàm lấy dử liệu bảng

public DataTable getCasi(){…}

public DataTable getCaSi\_by\_macasi(string macasi){…}

#endregion

#region các phương thức xử lý

public int themCaSi(string macasi, string tencasi){…}

public int xoaCaSi(string macasi){…}

public int capnhatCaSi(string macasi, string tencasi){…}

#endregion

}

### Lớp tác giả

Gồm các phương thức lấy bảng dữ liệu tác giả theo mã tác giả và các phương thức thêm, sửa, xóa tác giả.

Thực thi:

class Tacgia\_Data

{

DataConection objCon = new DataConection();

#region các hàm lấy bảng, lấy hết bảng, lấy theo điều kiện

public DataTable getTacGia(){…}

#endregion

#region các phương thức xử lý thêm xóa sữa (sql)

public int themTacGia(string matacgia, string tentacgia){…}

public int xoaTacGia(string matacgia){…}

public int capnhatTacGia(string matacgia, string tentacgia){…}

#endregion

}

### Lớp ca sĩ - tác giả

Gồm các phương thức lấy bài hát theo mã bài hát, lấy bài hát theo mã tác giả.

Thực thi:

class Casi\_Baihat\_Data

{

DataConection objCon = new DataConection();

// khởi tạo

public Casi\_Baihat\_Data() { }

#region các hàm lấy dử liệu bảng

public DataTable getCasi\_BaiHat\_by\_mabaihat(string mabaihat){…}

public DataTable getCasi\_BaiHat\_by\_macasi(string macasi){…}

#endregion

}

### Lớp hãng sản xuất

Gồm các phương thức lấy bảng dữ liệu hãng sản xuất theo mã hãng sản xuất và các phương thức thêm, sửa, xóa thông tin hãng sản xuất.

Thực thi:

class Hangsanxuat\_Data

{

DataConection objCon = new DataConection();

#region hàm lấy bảng NHASANXUAT

public DataTable getNhaSanXuat(){…}

#endregion

#region các phương thức xử lý

public int themNhaSX(string manhasx, string tennhasx){ …}

public int xoaNhaSanXuat(string manhasx){...}

public int capnhatNhaSX(string manhasx, string tennhasx){…}

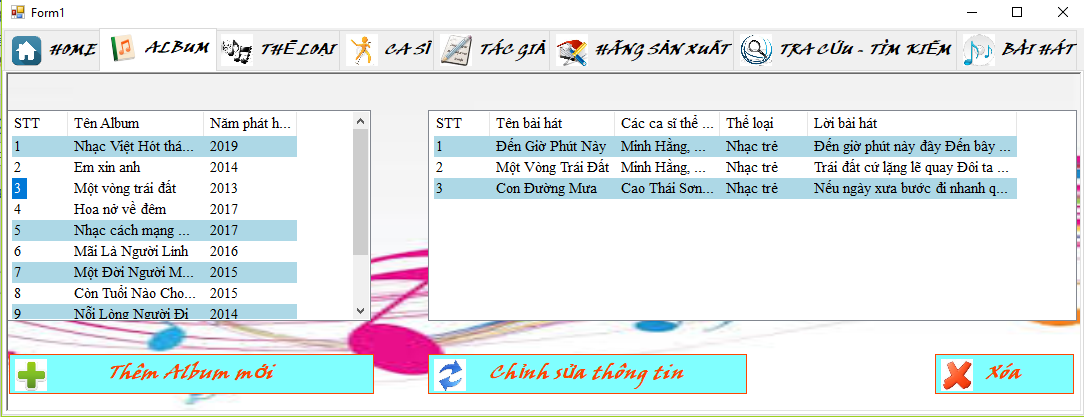
#endregion

}

# KẾT QUẢ CHẠY CHƯƠNG TRÌNH



Hình 16-Kết quả Load page home



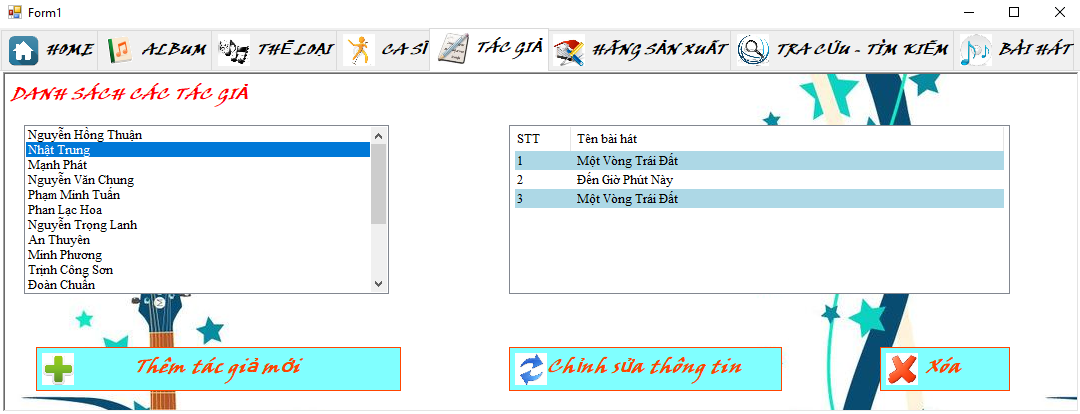
Hình 17- Kết quả load page ALBUM



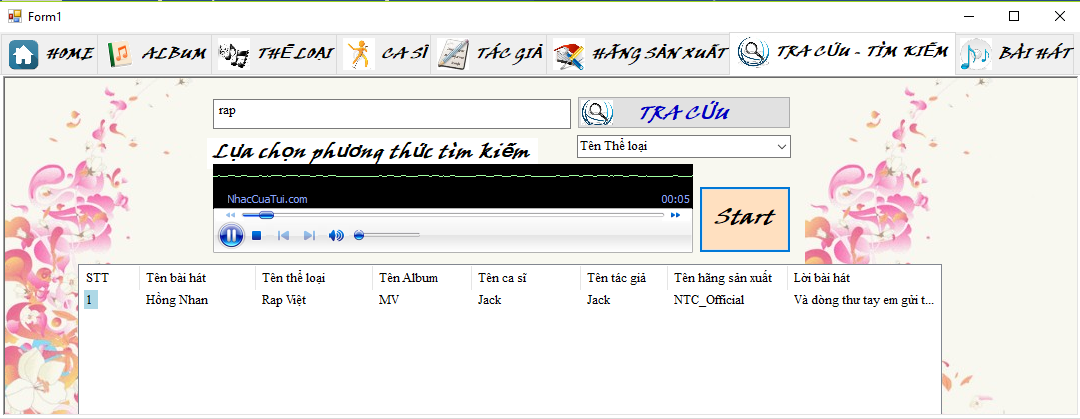
Hình 18- Kết quả load page THỂ LOẠI



Hình 19- Kết quả load page CA SĨ



Hình 20- Kết quả load page TÁC GIẢ



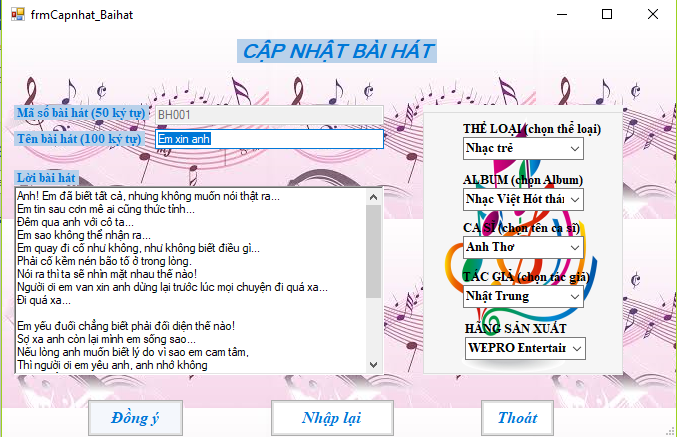
Hình 21- Kết quả load page TRA CỨU – TÌM KIẾM



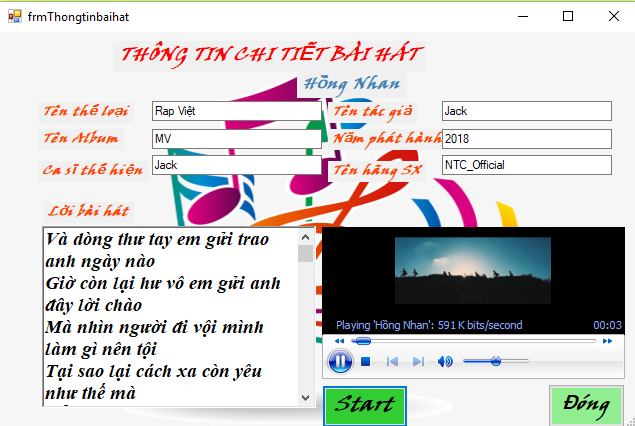
Hình 22- Kết quả load page BÀI HÁT



Hình 23- Kết quả load Form thêm bài hát mới



Hình 24-Kết quả load form cập nhật bài hát



Hình 25- Kết quả load Form chi tiết bài hát

# KẾT LUẬN

Công việc nhóm đã làm được:

* + Xây dựng được giao diện chương trình phù hợp với bài toán cho phép người dùng xem thông tin bài hát, thực hiện các thao tác thêm, sửa, xóa thông tin của bài hát.
  + Cho phép người dùng chơi nhạc thông qua giao diện chương trình.
  + Hoàn thành được cơ bản các yêu cầu bài tập lớn đã yêu cầu.

Những việc chưa làm được:

* + Code còn dài nhiều chức năng còn hạn chế.
  + Chưa tự động import được dữ liệu từ các nguồn trên mạng.
  + Chưa thêm được nhiều ca sĩ cùng hát một bài hát qua giao diện chương trình

Hướng phát triển:

* Từ những gì nhóm làm được, chúng ta có thể phát triển lên thành các phần mềm quản lý bài hát cho các quản karaoke.
* Thiết kế giao diện đẹp hơn, nhiều chức năng hơn.

# TÀI LIỆU THAM KHẢO

* *Bài giảng Lập trình Windows của Khoa công nghệ thông tin Trường đại học Hàng hải Việt Nam.*
* *Giáo trình của Howkteam.com*
* *Tài liệu của Linkerpt.com*