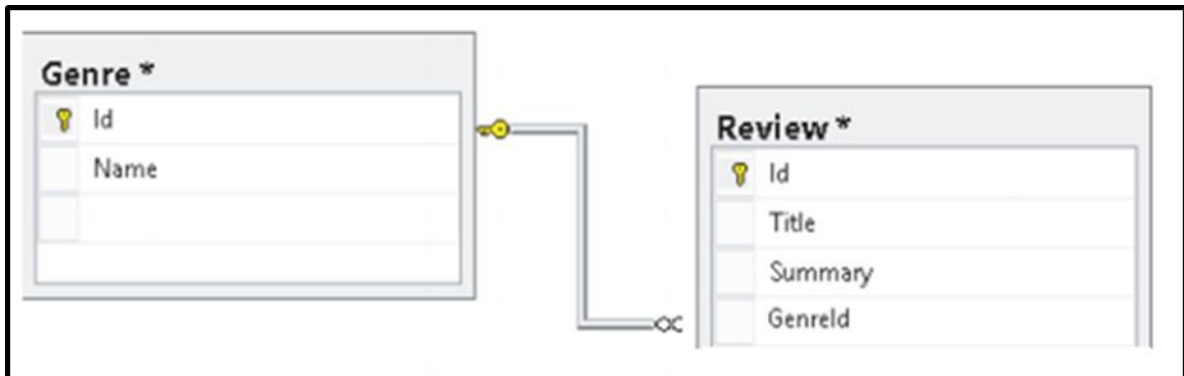


Tài liệu hướng dẫn thực hành

Tài liệu này sẽ hướng dẫn cách làm việc với các đối tượng DataSet, DataTable, DataRow, DataColumn và DataRelation thông qua việc xây dựng các ứng dụng trong môi trường Visual Studio. Hai bảng mẫu có lược đồ như sau:



Hai bảng được lưu trong một DataSet với ràng buộc phù hợp và cuối cùng là hiển thị dữ liệu hai bảng ra hai DataGridView.

Một vài lưu ý trước khi thực hành:

- Tài liệu sử dụng Visual Studio 2019 Community. Các phiên bản Visual Studio trước đó các bước thực hiện tương tự.
- Có thể xem lại kiến thức cơ bản về WPF tại <https://ngocminhtran.com/lam-quen-ung-dung-wpf/> (Mặc dù dùng VS 2012 nhưng các thành phần cơ bản trong WPF vẫn tương tự).

Bài thực hành: Sử dụng DataSet, DataTable, DataRelation

Bước 1: Mở Visual Studio 2019 Community và tạo một ứng dụng WPF App (.NET Framework) tên *UngDungDataSet*. Đổi tên cửa sổ giao diện thiết kế thành *Ứng Dụng DataSet*.

Bước 2: Trước khi tạo các bảng chúng ta sẽ định nghĩa một biến lớp kiểu DataSet đến lớp MainWindow phía trên hàm khởi tạo MainWindow():

```
private System.Data.DataSet dataSet;
public MainWindow()
{
    InitializeComponent();
}
```

Bước 3: Tạo hàm tên Tao_bang_Genre()

```
private void Tao_bang_Genre(){
```

```
}
```

Hàm này chứa hai đoạn mã:

Định nghĩa bảng Genre gồm hai cột là Id và Name, với Id là khóa chính của bảng như sau:

```
// Tạo một bảng mới tên Genre
System.Data.DataTable gen = new DataTable("Genre");
// khai báo các biến để tạo cột và hàng cho bảng
DataColumn column;
DataRow row;
// Tạo cột Id với kiểu dữ liệu
column = new DataColumn();
column.DataType = System.Type.GetType("System.String");
column.ColumnName = "Id";
column.ReadOnly = true;
column.Unique = true;
// thêm cột đến bảng
gen.Columns.Add(column);
// Tạo cột Name với kiểu dữ liệu
column = new DataColumn();
column.DataType = System.Type.GetType("System.String");
column.ColumnName = "Name";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
// thêm cột đến bảng
gen.Columns.Add(column);
// thiết lập Id là khóa chính
gen.PrimaryKey = new DataColumn[] { gen.Columns["Id"] };
```

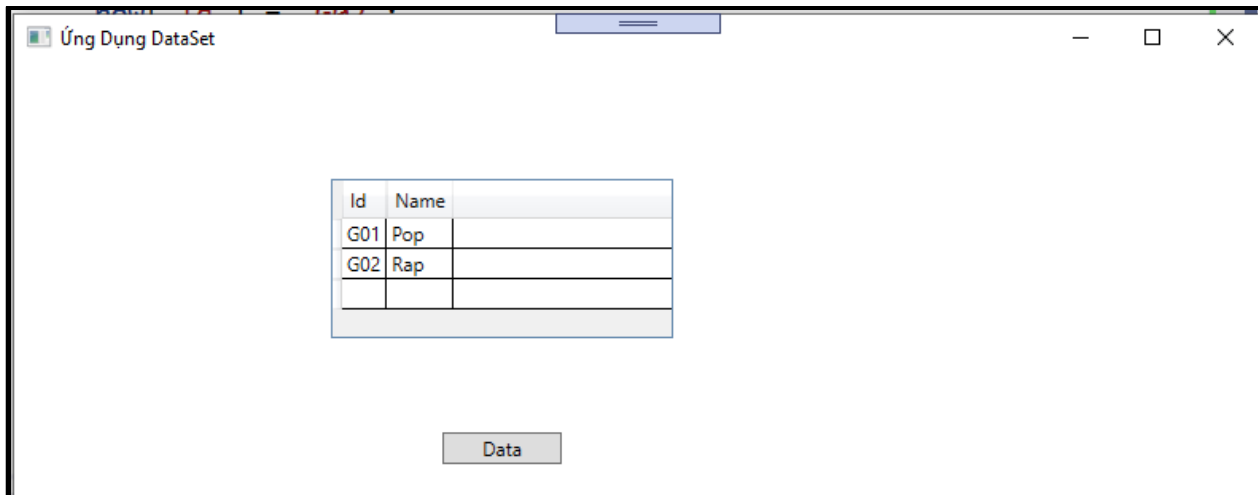
Thêm bảng Genre đến DataSet và tạo hai hàng dữ liệu cho bảng Genre:

```
// Tạo một DataSet
dataSet = new DataSet("Album");
// Thêm bảng Genre đến DataSet
dataSet.Tables.Add(gen);
//thêm 2 hàng đến bảng Genre
//thêm hàng thứ nhất
row = gen.NewRow();
row["Id"] = "G01";
row["Name"] = "Pop";
gen.Rows.Add(row);
//thêm hàng thứ hai
row = gen.NewRow();
row["Id"] = "G02";
row["Name"] = "Rap";
gen.Rows.Add(row);
```

Bước 4: Thêm một Button với thuộc tính Name là btnData và thuộc tính Content là Data đến giao diện. Kế đó, thêm một DataGrid với thuộc tính Name là dataGen đến giao diện. Thêm hàm xử lý sự kiện Click cho Button Data tên btnData_Click và thêm đoạn mã sau đến hàm này:

```
private void btnData_Click(object sender, RoutedEventArgs e)
{
    Tao_bang_Genre();
    dataGenre.ItemsSource = dataSet.Tables["Genre"].DefaultView;
}
```

Bước 5: Lưu tất cả và thực thi ứng dụng. Kết quả sau khi nhấn nút Data:



Dữ liệu từ bảng Genre hiển thị trong DataGrid sau khi nhấn nút Data

Đóng ứng dụng và chuyển sang tạo bảng Review.

Bước 6: Tương tự các bước tạo bảng Genre, chúng ta định nghĩa một hàm tên Tao_bang_Review() để tạo bảng Review như sau:

```
private void Tao_bang_Review()
{
    // Tạo một bảng mới tên Genre
    System.Data.DataTable review = new DataTable("Review");
    // khai báo các biến để tạo cột và hàng cho bảng
    DataColumn column;
    DataRow row;
    // Tạo cột Id với kiểu dữ liệu
    column = new DataColumn();
    column.DataType = System.Type.GetType("System.String");
    column.ColumnName = "Id";
    column.ReadOnly = true;
    column.Unique = true;
    // thêm cột đến bảng
    review.Columns.Add(column);

    // Tạo cột Title với kiểu dữ liệu
```

```
column = new DataColumn();
column.DataType = System.Type.GetType("System.String");
column.ColumnName = "Title";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
// thêm cột đến bảng
review.Columns.Add(column);

// Tạo cột Summary với kiểu dữ liệu
column = new DataColumn();
column.DataType = System.Type.GetType("System.String");
column.ColumnName = "Summary";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
// thêm cột đến bảng
review.Columns.Add(column);

// Tạo cột GenreId với kiểu dữ liệu
// Giá trị cột này chính là giá trị cột Id của bảng Genre
column = new DataColumn();
column.DataType = System.Type.GetType("System.String");
column.ColumnName = "GenreId";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
// thêm cột đến bảng
review.Columns.Add(column);

// thiết lập Id là khóa chính
review.PrimaryKey = new DataColumn[] { review.Columns["Id"] };

// Thêm bảng Genre đến DataSet
dataSet.Tables.Add(review);

//Tạo 2 hàng review có GenreId = G01
for (int i = 0; i < 2; i++)
{
    row = review.NewRow();
    row["Id"] = "R0" + i;
    row["Title"] = "Title" + i;
    row["Summary"] = "Summary" + i;
    row["GenreId"] = "G01";
    review.Rows.Add(row);
}
//Tạo 2 hàng review có GenreId = G02

for (int i = 0; i < 2; i++)
{
```

```

        row = review.NewRow();
        row["Id"] = "R0" + (i + 2);
        row["Title"] = "Title" + (i + 2);
        row["Summary"] = "Summary" + (i + 2);
        row["GenreId"] = "G02";
        review.Rows.Add(row);
    }
}

```

Bước 7: Trước khi hiển thị dữ liệu từ hai bảng ra hai DataGrid, chúng ta thiết lập quan hệ hai bảng Genre và Review dùng đối tượng DataRelation bằng cách tạo hàm Quan_he() như sau:

```

private void Quan_he()
{
    // DataRelation yêu cầu hai cột quan hệ với nhau.
    // Một cột là khóa chính (Id) ở bảng cha (Genre), một cột là khóa
    // ngoại (GenreId) ở bảng con (Review).
    // khóa chính
    DataColumn parentColumn =
        dataSet.Tables["Genre"].Columns["Id"];
    // khóa ngoại
    DataColumn childColumn =
        dataSet.Tables["Review"].Columns["GenreId"];
    //Đặt tên quan hệ
    DataRelation relation = new
        DataRelation("Gen_Review", parentColumn, childColumn);
    dataSet.Tables["Review"].ParentRelations.Add(relation);
}

```

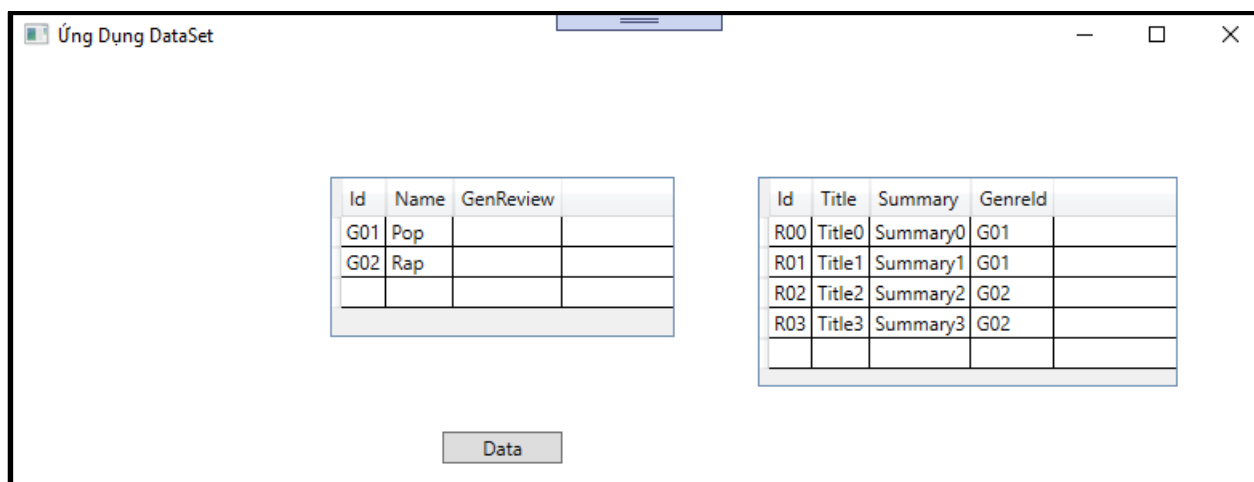
Bước 8: Thêm một DataGrid với thuộc tính Name là dataReview đến giao diện. Hàm xử lý sự kiện nút Data được viết lại như sau:

```

private void btnData_Click(object sender, RoutedEventArgs e)
{
    Tao_bang_Genre();
    Tao_bang_Review();
    Quan_he();
    dataGenre.ItemsSource = dataSet.Tables["Genre"].DefaultView;
    dataReview.ItemsSource = dataSet.Tables["Review"].DefaultView;
}

```

Bước 9: Lưu tất cả và thực thi ứng dụng. Kết quả sau khi nhấn nút Data:



Dữ liệu từ bảng Genre và Review hiển thị trong hai DataGrid sau khi nhấn nút Data
Đóng ứng dụng.

Kết thúc bài thực hành.