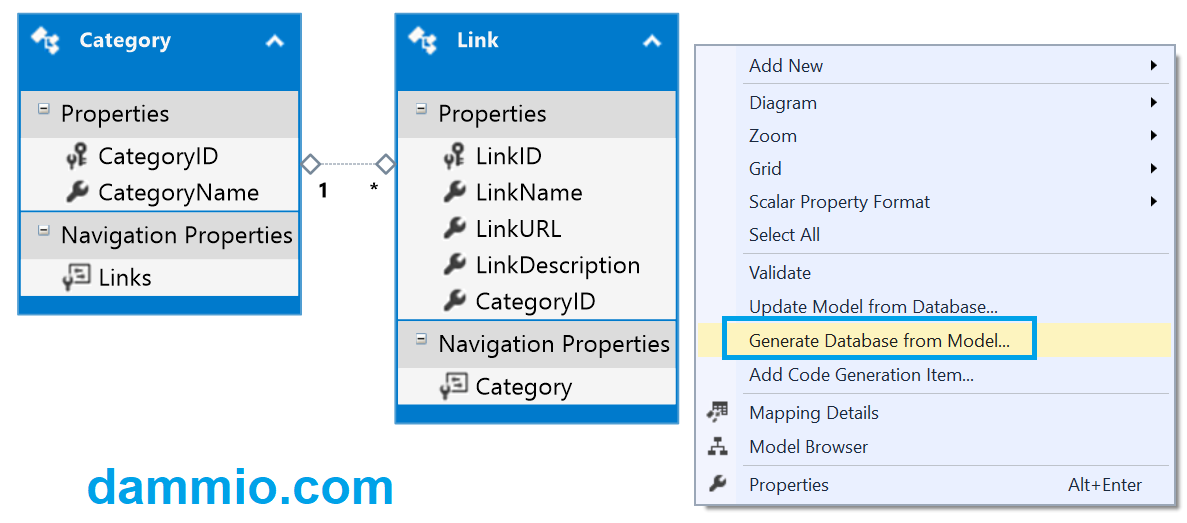
Phần 11: Thêm tìm kiếm (mở rộng)

Như đã hướng dẫn ở bài viết phần 10 về cách tìm kiếm liên kết theo tiêu đề (Title) và Danh mục. Bài viết này sẽ hướng dẫn bạn cách tìm kiếm mà bảng Link và bảng Category chứa các thuộc tính điều hướng (**navigation links**). Điều bắt buộc để tìm kiếm theo thuộc tính điều hướng là bạn phải tạo database hoặc mô hình có liên kết rằng buộc với nhau. Bạn có thể xem lại mô hình database như sau:



Trong mô hình trên, bạn có thể thấy bảng Category và bảng Link quan hệ với nhau theo kiểu 1-n, tức là 1 danh mục (Category) có chứa nhiều liên kết (Link), hay 1 liên kết (Link) chỉ thuộc về 1 danh mục (Category) duy nhất.

# Tạo cơ sở dữ liệu

Mở **Microsoft SQL Server**, bạn tạo mới cơ sở dữ liệu tên là **DammioMVC1** (để tránh trùng với DammioMVC ở Phần 10). Bạn chọn nút New Query để tạo một tập tin query mới làm chỗ để paste đoạn code sau.

USE [DammioMVC1]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Category] Script Date: 10/23/2018 16:34:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Category](

[CategoryID] [int] IDENTITY(1,1) NOT NULL,

[CategoryName] [nvarchar](50) NULL,

CONSTRAINT [PK\_Category] PRIMARY KEY CLUSTERED

(

[CategoryID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Link] Script Date: 10/23/2018 16:34:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Link](

[LinkID] [int] IDENTITY(1,1) NOT NULL,

[LinkName] [nvarchar](50) NULL,

[LinkURL] [nvarchar](50) NULL,

[LinkDescription] [nvarchar](50) NULL,

[CategoryID] [int] NULL,

CONSTRAINT [PK\_Link] PRIMARY KEY CLUSTERED

(

[LinkID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: ForeignKey [FK\_Link\_Category] Script Date: 10/23/2018 16:34:46 \*\*\*\*\*\*/

ALTER TABLE [dbo].[Link] WITH CHECK ADD CONSTRAINT [FK\_Link\_Category] FOREIGN KEY([CategoryID])

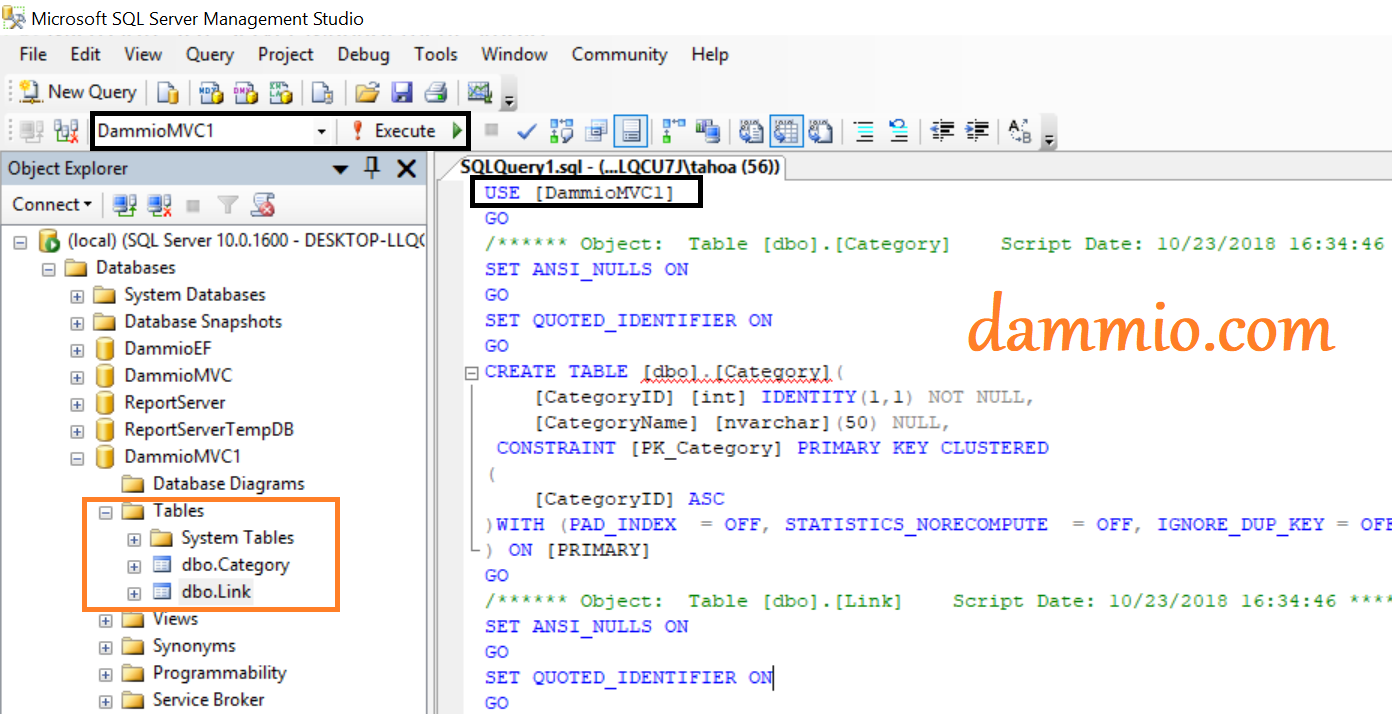
REFERENCES [dbo].[Category] ([CategoryID])

GO

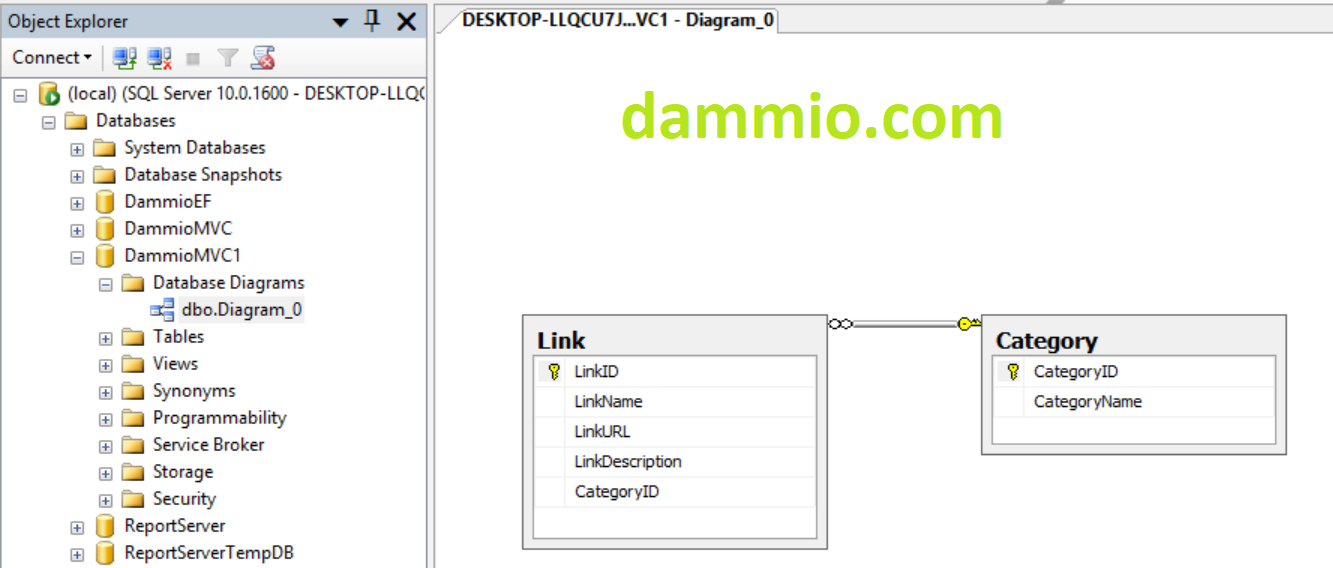
ALTER TABLE [dbo].[Link] CHECK CONSTRAINT [FK\_Link\_Category]

GO

Sau đó, bạn nhấn nút Execute để thực thi đoạn mã truy vấn này trên database.



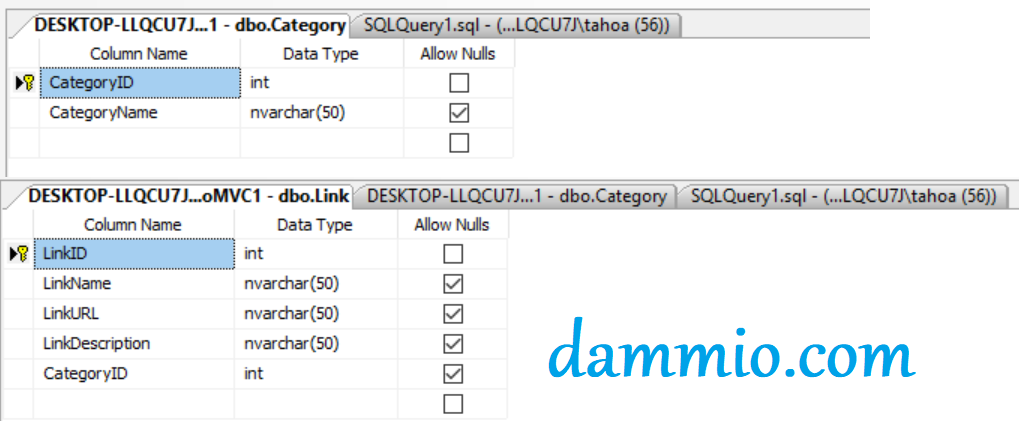
Sau đó, ở phần Diagram trong tạo 1 sơ đồ kết nối giữa Category và Link dựa trên khóa ngoại CategoryID theo hình sau.



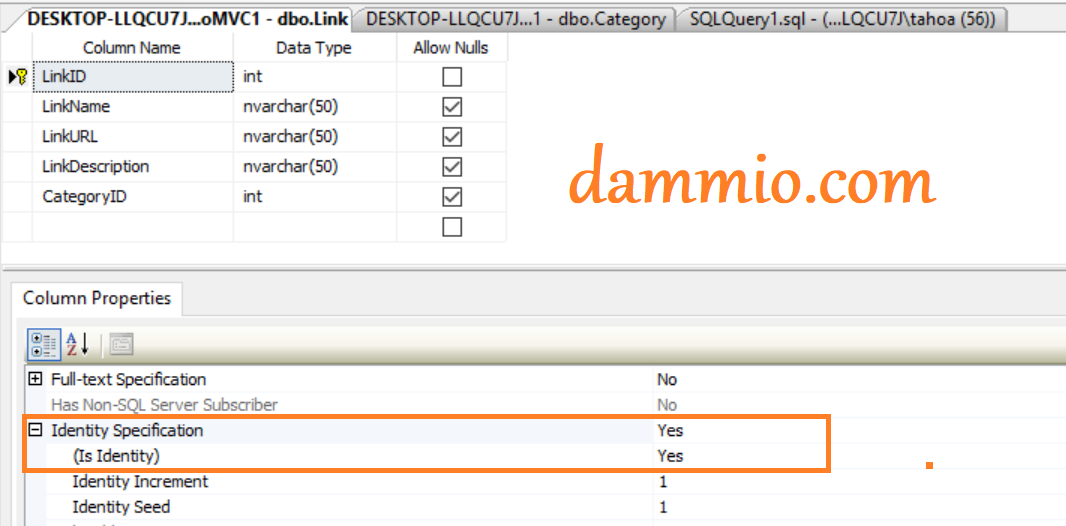
# Khóa tự tăng

Phần này chỉ là kiến thức mỏ rộng, bạn không cần làm gì cả ở phần này.

Sau khi database tạo xong, bạn có thể thấy xuất hiện 2 bảng Category và Link ở database. Ở bước này, mình hướng dẫn về cấu trúc các bảng trong database cho các bạn không theo dõi từ đầu. Chúng ta có 2 bảng Category và Link với khóa tự tăng kiểu int có cấu trúc.

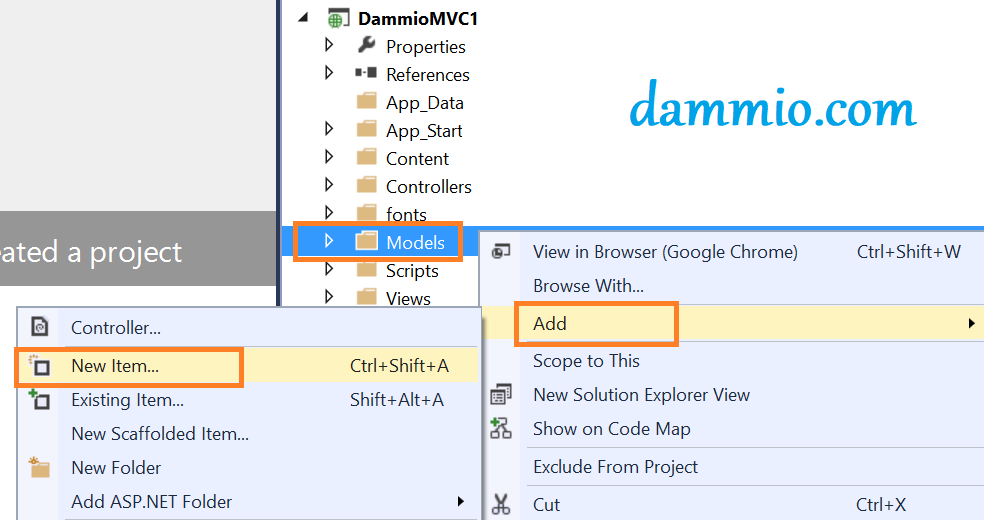


Nhiều bạn là “tay ngang” xem phần này sẽ tự hỏi làm sao để thiết kế database có khóa tự tăng thì rất đơn giản, bạn chọn khóa kiểu int và định nghĩa thuộc tính Identify Specification là Yes như hình sau.

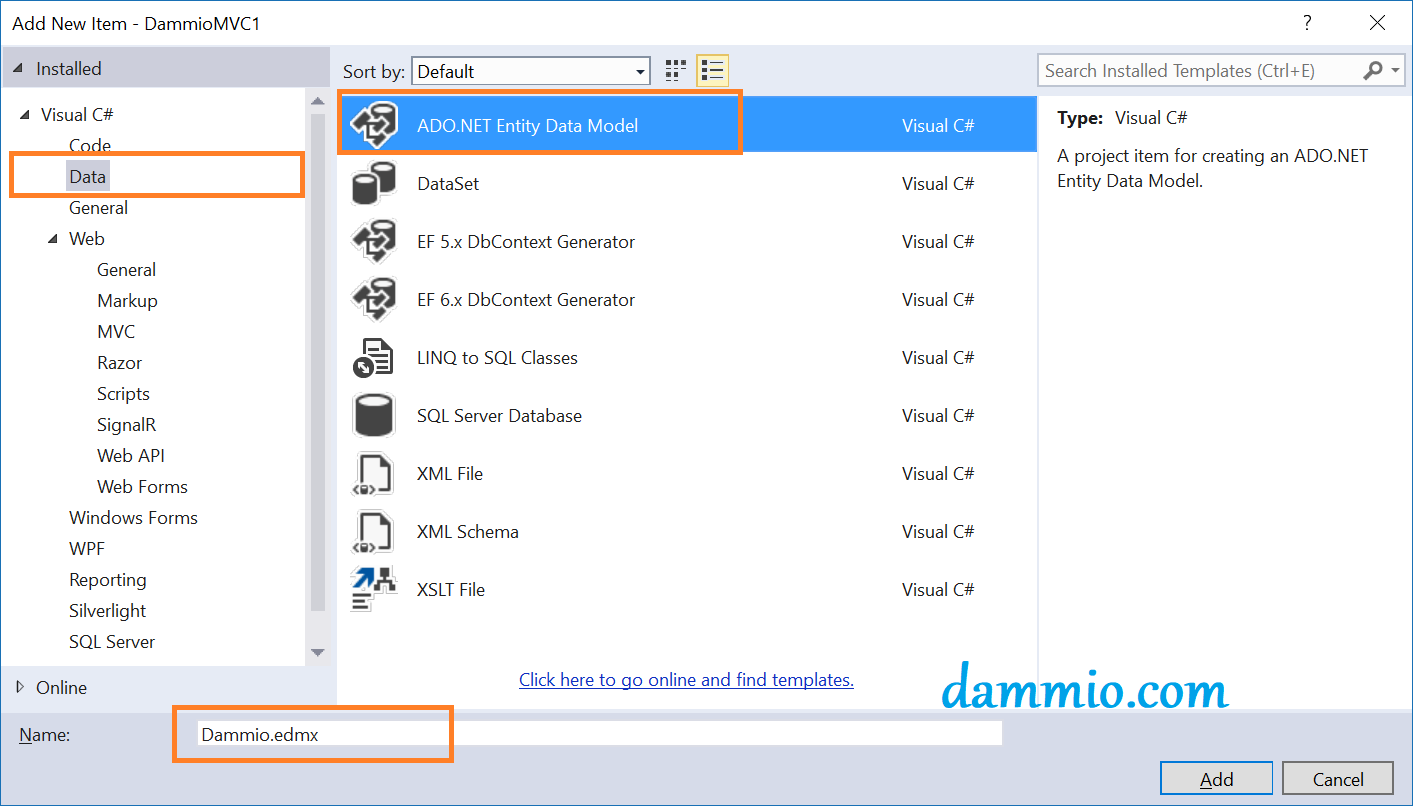


# Tạo dự án và gieo mô hình code

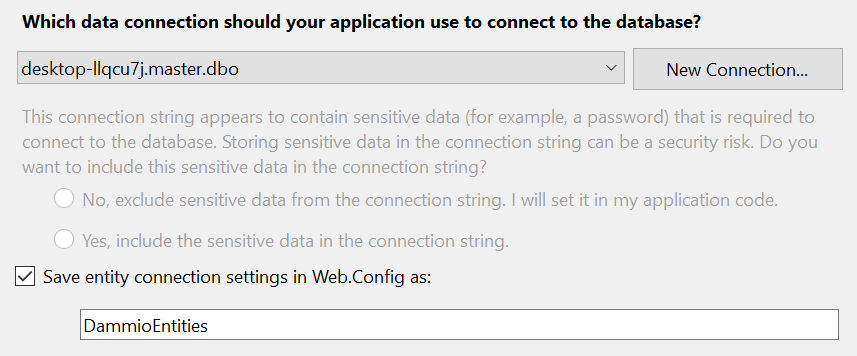
Sau khi tạo xong database, bạn mở Visual Studio tạo 1 dự án [MVC](https://www.dammio.com/glossary/mvc) mới tên là **DammioMVC1**. Sau đó, chọn click chuột vào thư mục Models, nhấp phải chọn Add -> New Item.



Sau đó, khi cửa sổ Add New Item hiện ra, bạn chọn ADO.NET Entity Data Model, đặt tên mô hình database là dammio.edmx và nhấn Add.

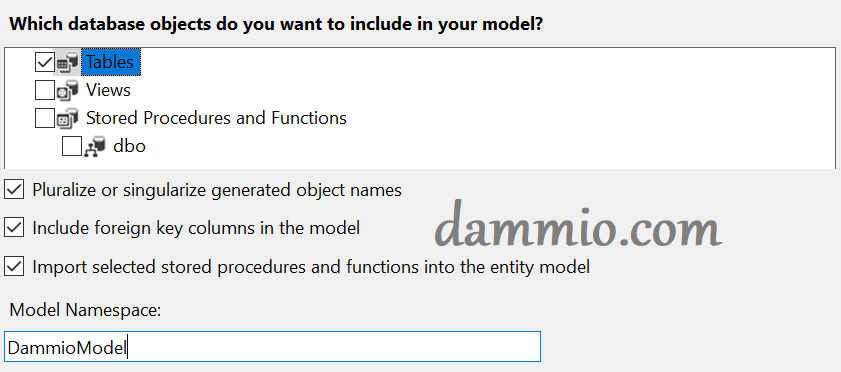


Kế đến bạn chọn chế độ **Generate from database** để gieo mã nguồn từ database, sau đó đặt tên cho kết nối database là **DammioEntities**.



Tiếp theo, chọn Tables và đặt tên namespace mô hình là DammioModel, nhấn Finish để hoàn thành việc gieo mô hình.

Bạn nên bỏ dấu checkbox cuối cùng “Import selected stored procedures and functions…” nếu không dùng SP trong database.



Sau khi gieo mô hình xong, bạn mở nội dung tập tin **Models/Category.cs và Models/Link.cs** nếu thấy các thuộc tính virtual sau xuất hiện ở mã nguồn 2 lớp này là được.

namespace DammioMVC1.Models

{

using System;

using System.Collections.Generic;

public partial class Category

{

public Category()

{

this.Links = new HashSet<Link>();

}

public int CategoryID { get; set; }

public string CategoryName { get; set; }

// Thuộc tính liên kết với bảng Link

public virtual ICollection<Link> Links { get; set; }

}

}

namespace DammioMVC1.Models

{

using System;

using System.Collections.Generic;

public partial class Link

{

public int LinkID { get; set; }

public string LinkName { get; set; }

public string LinkURL { get; set; }

public string LinkDescription { get; set; }

public Nullable<int> CategoryID { get; set; }

// Thuộc tính liên kết với bảng Category

public virtual Category Category { get; set; }

}

}

Đến đây, bạn gieo lại Controller và Views cho bảng Link và Category như các bài [**Phần 7: Truy cập mô hình dữ liệu từ Controller**](https://www.dammio.com/2018/07/01/asp-net-mvc-phan-7-truy-cap-mo-hinh-du-lieu-tu-controller). Sau đó chạy [URL](https://www.dammio.com/glossary/url) http://localhost:xxxx/Link và http://localhost:xxxx/Category để thêm một số dữ liệu demo.

# Sửa tập tin Controllers/LinkController.cs

Tiếp theo, bạn mở tập tin **Controllers/LinkController.cs**, tìm hàm Index và sửa như sau:

public ActionResult Index(string searchString, int categoryID = 0)

{

//1. Tạo danh sách danh mục để hiển thị ở giao diện View thông qua DropDownList

var categories = from c in db.Categories select c;

ViewBag.categoryID = new SelectList(categories, "CategoryID", "CategoryName"); // danh sách Category

//2. Tạo câu truy vấn kết 2 bảng Link, Category bằng hàm Include do 2 bảng có ràng buộc ở các thuộc tính virtual

var links = db.Links.Include(l => l.Category);

//3. Tìm kiếm chuỗi truy vấn

if (!String.IsNullOrEmpty(searchString))

{

links = links.Where(s => s.LinkName.Contains(searchString));

}

//4. Tìm kiếm theo CategoryID

if (categoryID != 0)

{

links = links.Where(x => x.CategoryID == categoryID);

}

return View(links.ToList());

}

Bạn chỉ cần chú ý mục số 2, theo đó đoạn mã **var links = db.Links.Include(l => l.Category);** dùng hàm **Include** cho phép bạn lấy tất cả Link có kèm theo Category thay vì phải dùng mệnh đề join như [**Phần 10**](https://www.dammio.com/2018/07/06/asp-net-mvc-phan-10-them-tim-kiem). Đây là lợi thế khi bạn tạo một cơ sở dữ liệu có mô hình ràng buộc với nhau.

# Sửa tập tin View/Link/Index.cshtml

Bạn mở tập tin này để chúng ta sửa giao diện tìm kiếm như sau. Bạn chỉ cần thêm vào đoạn cần thêm như mô tả trong code.

@model IEnumerable<DammioMVC1.Models.Link>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

@Html.ActionLink("Create New", "Create")

<!-- 1. Đoạn cần thêm -->

@using (Html.BeginForm("Index", "Link", FormMethod.Get))

{

<p>

Title: @Html.TextBox("SearchString") --- Category: @Html.DropDownList("categoryID", "All") <input type="submit" value="Tìm kiếm" />

</p>

}

<!-- Kết thúc -->

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.LinkName)

</th>

<th>

@Html.DisplayNameFor(model => model.LinkURL)

</th>

<th>

@Html.DisplayNameFor(model => model.LinkDescription)

</th>

<th>

@Html.DisplayNameFor(model => model.Category.CategoryName)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.LinkName)

</td>

<td>

@Html.DisplayFor(modelItem => item.LinkURL)

</td>

<td>

@Html.DisplayFor(modelItem => item.LinkDescription)

</td>

<td>

@Html.DisplayFor(modelItem => item.Category.CategoryName)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id=item.LinkID }) |

@Html.ActionLink("Details", "Details", new { id=item.LinkID }) |

@Html.ActionLink("Delete", "Delete", new { id=item.LinkID })

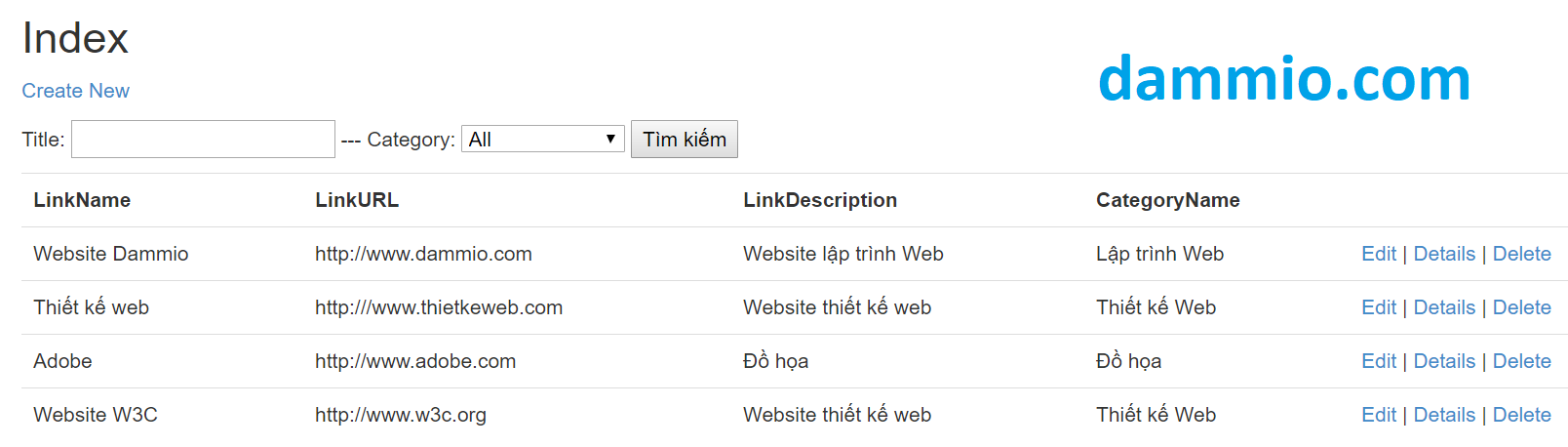
</td>

</tr>

}

</table>

Như vậy, thông qua biến categoryID (ViewBag), bạn có thể lấy tất cả danh mục tìm kiếm hiển thị ở DropDownList, sau đó người dùng sẽ nhập từ khóa (SearchString) và chọn danh mục tìm kiếm. Sau đó, các biến này truyền đến hàm Index của LinkController.cs để lấy dữ liệu truy vấn và trả về View để hiển thị kết quả tìm kiếm được.



1. **Kết luận**

Phần 11 là cách tìm kiếm bài bản hơn phần 10 khi mà bạn phải thiết lập database có sơ đồ ràng buộc (quan hệ) giữa các bảng thông qua khóa chính, khóa phụ. Từ đó, bạn có thể gieo mô hình và sử dụng **phương thức truy vấn Include** để lấy dữ liệu các bảng liên quan dễ dàng hơn thay vì các câu truy vấn dài dòng. Mời bạn tiếp tục theo dõi phần tiếp theo.

[1].[DammioMVC1\_phan11.rar](https://www.dammio.com/wp-content/uploads/2018/10/DammioMVC1_phan11.rar)