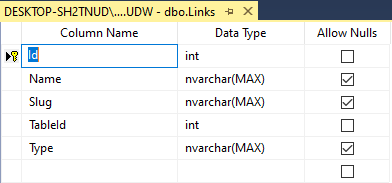
# Bài thực hành 7: xây dựng trang quản trị cho LINKs

Ta có thiết kế SQL của LINKS như sau:



## Chỉnh sửa nội dung Model\LINKS: Việt hóa

using System;

…

namespace MyClass.Model

{

[Table("Products")]

public class Products

{

[Key]

public int Id { get; set; }

[Display(Name = "Tên liên kết")]

public string Name { get; set; }

[Display(Name = "URL")]

public string Slug { get; set; }

[Required(ErrorMessage = "ID bảng liên kết không được để trống")]

[Display(Name = "Bảng liên kết")]

public int TableId { get; set; }

[Display(Name = "Loại liên kết")]

public string Type { get; set; }

}

}

# Tạo LinkSDAO từ CategoriesDAO

R\_Click MyClass - DAO - Add - Class

Name = LinkDAO

#### (Không cần tạo Controller)

#### 

#### Lưu ý: phần Index có thể không làm vì mục Link chúng ta không không tạo trang Index

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using MyClass.Model;

namespace MyClass.DAO

{

public class LinksDAO

{

private MyDBContext db = new MyDBContext();

/////////////////////////////////////////////////////////////////////////////////////

//Hien thi danh sach toan bo Loai san pham: SELCT \* FROM

public List<Links> getList(string status = "All")

{

List<Links> list = null;

return list;

}

/////////////////////////////////////////////////////////////////////////////////////

//Hien thi danh sach 1 mau tin (ban ghi)

public Links getRow(int tableid, string typelink)

{

return db.Links

.Where(m => m.TableId == tableid && m.Type == typelink)

.FirstOrDefault();

}

/////////////////////////////////////////////////////////////////////////////////////

///Them moi mot mau tin

public int Insert(Links row)

{

db.Links.Add(row);

return db.SaveChanges();

}

/////////////////////////////////////////////////////////////////////////////////////

///Cap nhat mot mau tin

public int Update(Links row)

{

db.Entry(row).State = EntityState.Modified;

return db.SaveChanges();

}

/////////////////////////////////////////////////////////////////////////////////////

///Xoa mot mau tin Xoa ra khoi CSDL

public int Delete(Links row)

{

db.Links.Remove(row);

return db.SaveChanges();

}

}

}

# Chỉnh sửa nội dung của các Controller có liên quan đến LINKS

## CategoriesController

### Gán các trường cho bảng Links: CREATE

/////////////////////////////////////////////////////////////////////////////////////

// POST: Admin/Category/Create: Them moi mot mau tin

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Categories categories)

{

if (ModelState.IsValid)

{

…

//xu ly cho muc Topics

if (categoryDAO.Insert(categories) == 1)//khi them du lieu thanh cong

{

Links links = new Links();

links.Slug = categories.Slug;

links.TableId = categories.Id;

links.Type = "category";[[1]](#footnote-1)

}

}

…

}

### Cập nhật bảng dữ liệu

#### Giống như cập nhật cho bảng Categories: categoryDAO.Insert(categories)

Trước tiên, cần tạo ra đối tượng linkDAO kế thừa từ LinksDAO

public class CategoryController : Controller

{

CategoriesDAO categoryDAO = new CategoriesDAO();

LinksDAO linksDAO = new LinksDAO();

…

//xu ly cho muc Topics

if (categoryDAO.Insert(categories) == 1)//khi them du lieu thanh cong

{

Links links = new Links();

links.Slug = categories.Slug;

links.TableId = categories.Id;

links.Type = "category";

linksDAO.Insert(links);[[2]](#footnote-2)

}

…

}

### Gán các trường cho bảng Links: EDIT

/////////////////////////////////////////////////////////////////////////////////////

// GET: Admin/Category/Edit/5: Cap nhat mau tin

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int? id)

{

…

//Cap nhat du lieu, sua them cho phan Links phuc vu cho Topics

//Neu trung khop thong tin: Type = category va TableID = categories.ID

Links links = linksDAO.getRow(categories.Id, "category");[[3]](#footnote-3)

if (categoryDAO.Update(categories)==1)

{

//Cap nhat du lieu

links.Slug = categories.Slug;

linksDAO.Update(links);

}

…

}

### Gán các trường cho bảng Links: DELETE

/////////////////////////////////////////////////////////////////////////////////////

// GET: Admin/Category/Delete/5: Cap nhat mau tin

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

…

//tim thay mau tin thi xoa, cap nhat cho Links

if (categoryDAO.Delete(categories) ==1)

{

Links links = linksDAO.getRow(categories.Id, "category");

//Xoa luon cho Links

linksDAO.Delete(links);

}

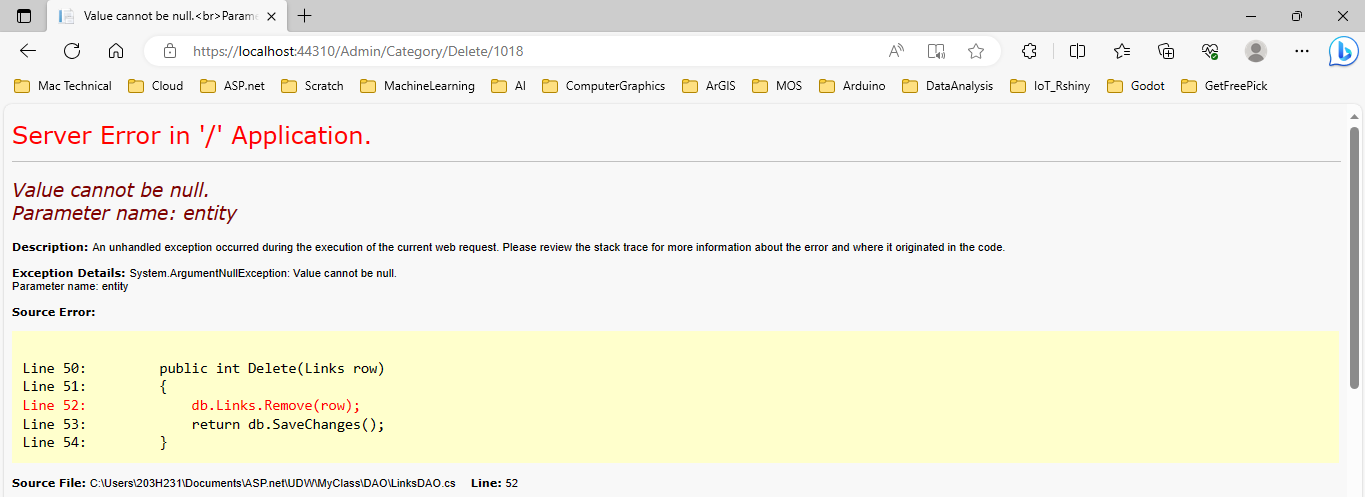
…

}

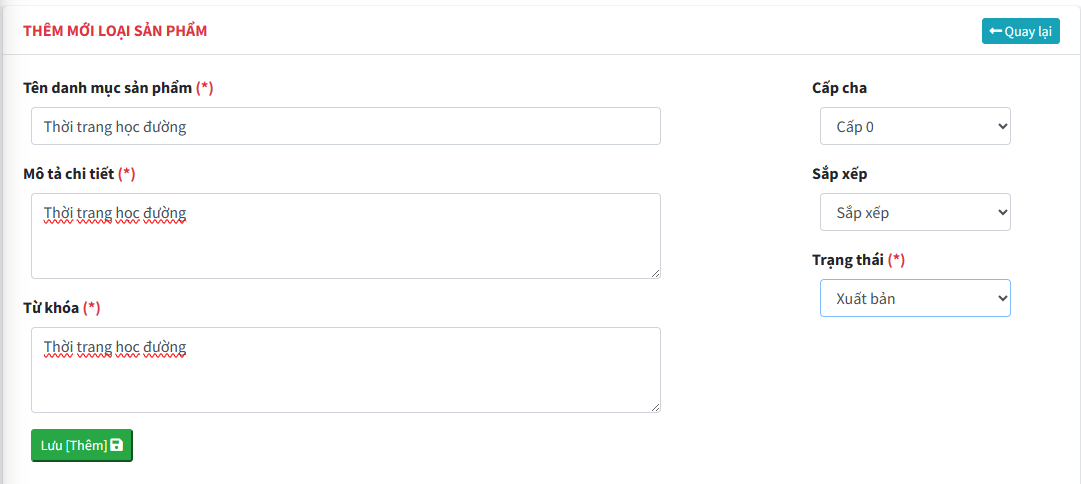
# Thử nghiệm

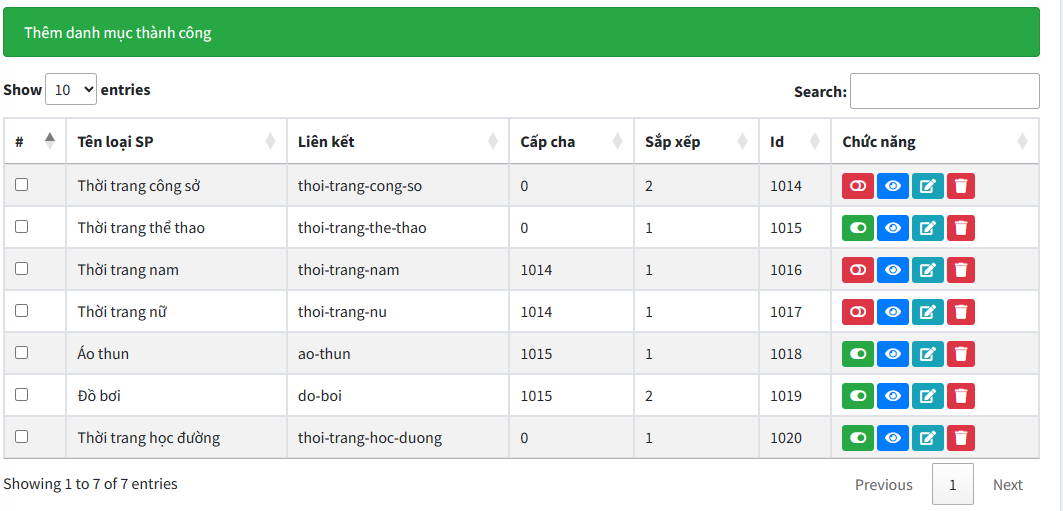
Trước khi thực hiện lưu ý:

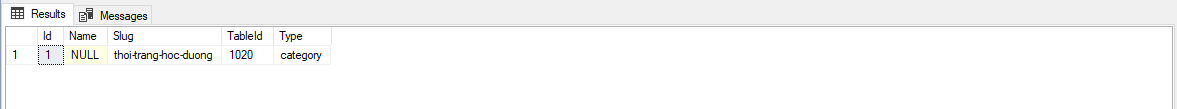
Những Categories đã tạo trước đây thì chưa có LINK, do vậy khi tiến hành DELETE sẽ lỗi:



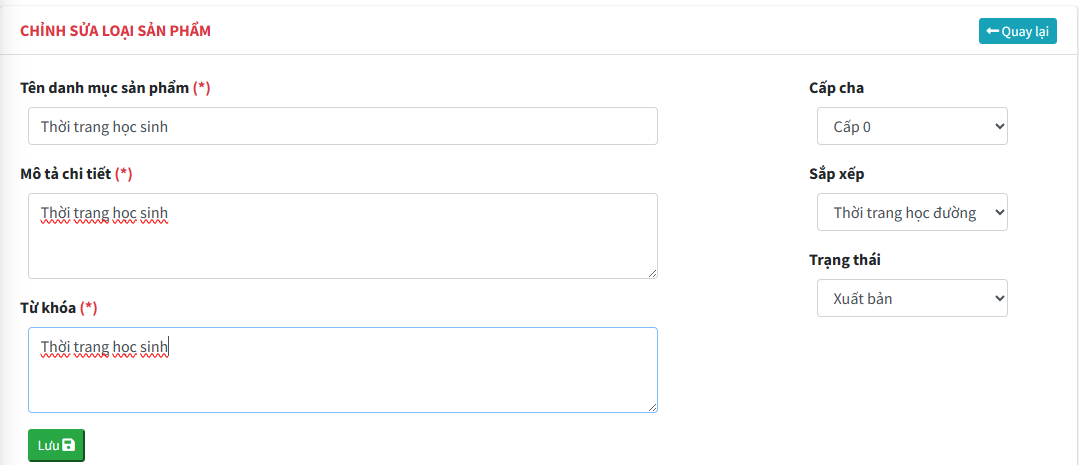
## TẠO MỚI CATEGORIES



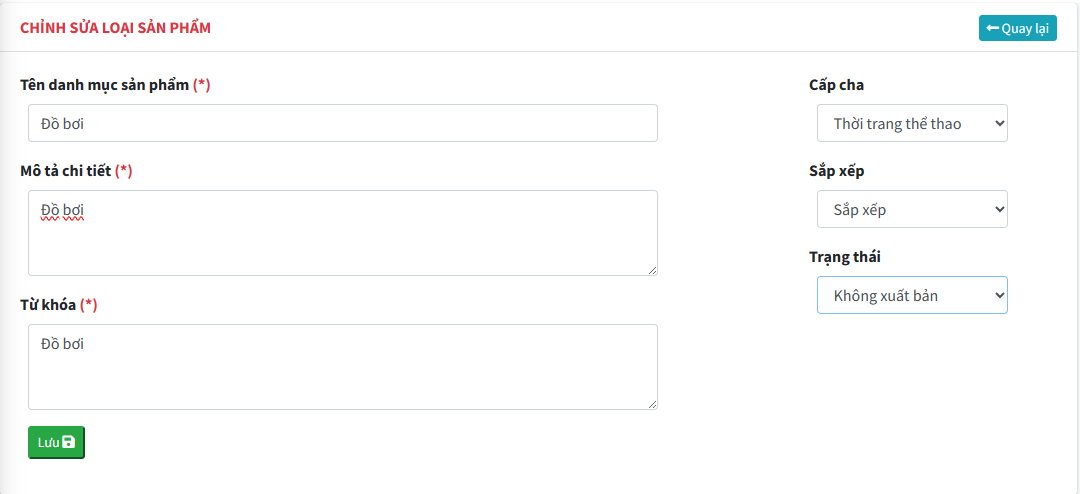


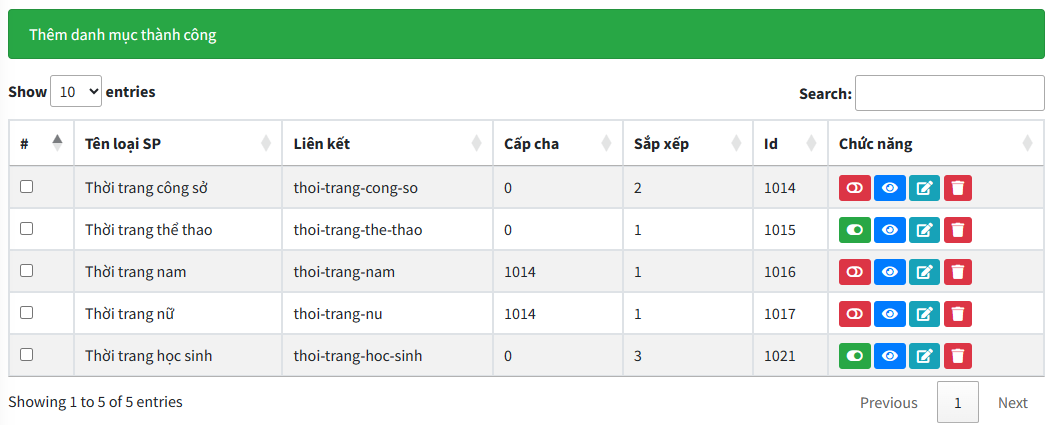


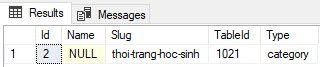
Những Categories mới sinh ra thì Edit và Delete bình thường



## EDIT

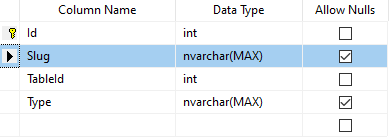






# Do vậy, hãy vào SQL và DELETE các Categories đã tạo trước đây

## Xóa trường Name ra khỏi Link



## Xóa trong Model

public class Links

{

[Key]

public int Id { get; set; }

[Display(Name = "URL")]

public string Slug { get; set; }

[Required(ErrorMessage = "ID bảng liên kết không được để trống")]

[Display(Name = "Bảng liên kết")]

public int TableId { get; set; }

[Display(Name = "Loại liên kết")]

public string Type { get; set; }

}

1. Link có 1 số kiểu: categories, supplier, products… ứng với mỗi loại khác nhau ta sẽ gán giá trị khác nhau [↑](#footnote-ref-1)
2. Chèn vào cơ sở dữ liệu [↑](#footnote-ref-2)
3. Xem hàm linksDAO.getRow với tham số đầu vào TablleID == CategoriesID và Type = “category” được khai báo ở hàm CREATE [↑](#footnote-ref-3)