

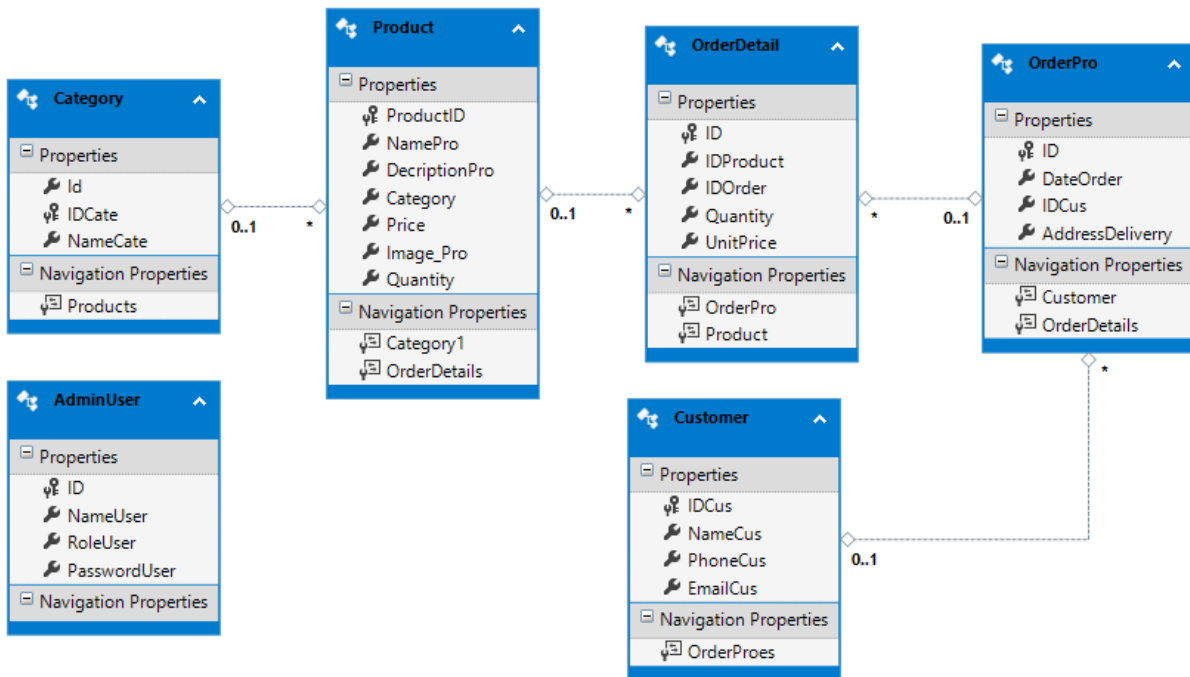
# THỰC HÀNH LẬP TRÌNH WEB NÂNG CAO

## BÀI THỰC HÀNH 08: Xử lý số lượng tồn trong bảng Product

### Kết nối CSDL bằng Database First

- Cơ sở dữ liệu được sử dụng trong bài thực hành: **DBSportStore**

[Link tải database](#)



- 1) Trong CSDL, trong bảng **Product** chắc chắn bạn có cột Quantity kiểu dữ liệu là **int**, nếu chưa có thì bổ sung

	Name	Data Type	Allow Nulls
PK	ProductID	int	<input type="checkbox"/>
	NamePro	nvarchar(MAX)	<input checked="" type="checkbox"/>
	DecriptionPro	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Category	nchar(20)	<input checked="" type="checkbox"/>
	Price	decimal(18,2)	<input checked="" type="checkbox"/>
	Image_Pro	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Quantity	int	<input checked="" type="checkbox"/>

## Model / Product.cs

```
namespace DemoConectDB.Models
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel.DataAnnotations.Schema;
    using System.Web;
    public partial class Product
    {
        public Product()
        {
            this.OrderDetails = new HashSet<OrderDetail>();
            Image_Pro = "~/Content/images/add.png";
        }
        public int ProductID { get; set; }
        public string NamePro { get; set; }
        public string DecriptionPro { get; set; }
        public string Category { get; set; }
        public Nullable<decimal> Price { get; set; }
        public string Image_Pro { get; set; }
        public Nullable<int> Quantity { get; set; }

        public virtual Category Category1 { get; set; }
        public virtual ICollection<OrderDetail> OrderDetails { get; set; }
        [NotMapped]
        public HttpPostedFileBase UploadImage { get; set; }
    }
}
```

**Bước 1:** Mở mở lớp **Model / Cart.cs** => thực hiện **update** lại hàm **Update\_quantity**

```
public void Update_quantity(int id, int _new_quan)
{
    var item = items.Find(s => s._product.ProductID == id );

    if (item != null)
    {
        if (items.Find(s => s._product.Quantity > _new_quan) != null) //nếu số lượng mua nhỏ hơn số lượng tồn
            item._quantity = _new_quan; //thì chấp nhận số lượng mua
        else item._quantity = 1; //ngược lại, thì số lượng mua trả về 1
    }
}
```

**Bước 2:** Mở **ShoppingCartController.cs** cập nhật lại hàm **CheckOut**

```
public ActionResult CheckOut(FormCollection form)
{
    try
    {
        Cart cart = Session["Cart"] as Cart;
        OrderPro _order = new OrderPro(); //Bang Hoa Don San pham
        _order.DateOrder = DateTime.Now;
        _order.AddressDeliverry = form["AddressDelivery"];
        _order.IDCus = int.Parse(form["CodeCustomer"]);
        database.OrderProes.Add(_order);
        foreach (var item in cart.Items)
        {
            OrderDetail _order_detail = new OrderDetail(); //Luu dong san pham vao bang Chi tiet Hoa don
            _order_detail.IDOrder = _order.ID;
            _order_detail.IDProduct = item._product.ProductID;
            _order_detail.UnitPrice =(double) item._product.Price;
            _order_detail.Quantity = item._quantity;
            database.OrderDetails.Add(_order_detail);
            //xử lý cập nhật lại số lượng tồn trong bảng Product
            foreach (var p in database.Products.Where(s=>s.ProductID==_order_detail.IDProduct)) //lấy ID Product đang có trong giỏ hàng
            {
                var update_quan_pro = p.Quantity - item._quantity; //số lượng tồn mới = số lượng tồn - số lượng đã mua
                p.Quantity = update_quan_pro; //thực hiện cập nhật lại số lượng tồn cho cột Quantity của bảng Product
            }
        }
        database.SaveChanges();
        cart.ClearCart();
        return RedirectToAction("Checkout_Success", "ShoppingCart");
    }
    catch
    {
        return Content("Error checkout. Please check information of Customer...Thanks.");
    }
}
```

