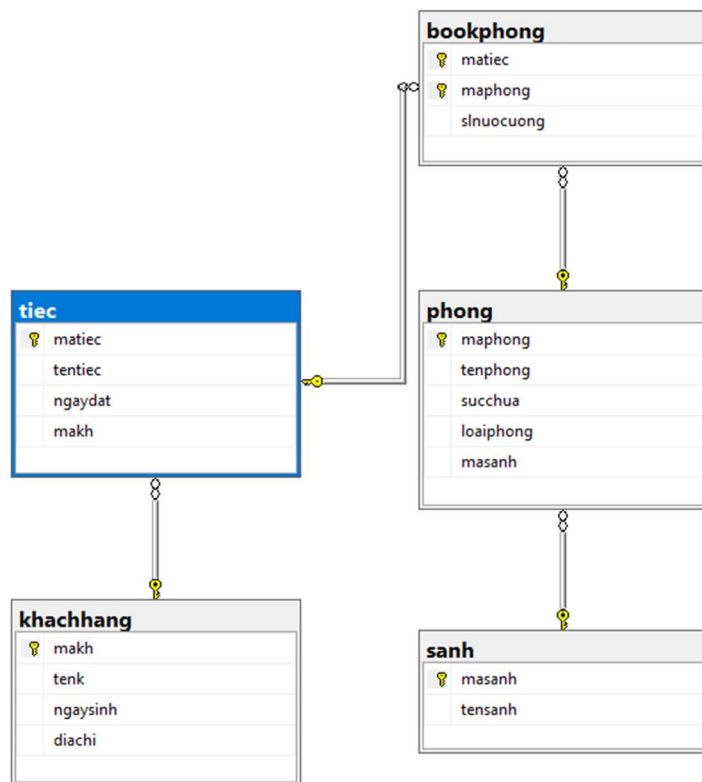


CHỦ ĐỀ 1



Model trong ASP.NET Core



A. CƠ SỞ DỮ LIỆU CHUẨN BỊ



- Datatype được lưu trong file “QLNH Script”
- Trước khi cài đặt cần phải tạo database QLNH trước.
- Chuẩn bị 3 gói Nuget

	Microsoft.EntityFrameworkCore.Design by Microsoft	3.1.20
	Shared design-time components for Entity Framework Core tools.	5.0.11
	Microsoft.EntityFrameworkCore.SqlServer by Microsoft	3.1.20
	Microsoft SQL Server database provider for Entity Framework Core.	5.0.11
	Microsoft.EntityFrameworkCore.Tools by Microsoft	3.1.20
	Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.	5.0.11



B. THỰC HIỆN TẠO MODEL THỦ CÔNG

Bước 1: Sinh viên tạo các model có sẵn trong

- Tạo Model Sảnh

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace ProjectTraining.Models
{
    public class SanhModel
    {
        15 references
        public string maSanh { get; set; }

        11 references
        public string tenSanh { get; set; }

        0 references
        public ICollection<PhongModel> Phong { get; set; }
    }
}
```

- Tạo Model Phòng

```
21 references
public class PhongModel
{
    15 references
    public string maPhong { get; set; }

    12 references
    public string tenPhong { get; set; }

    12 references
    public int sucChua { get; set; }

    12 references
    public string loaiPhong { get; set; }

    12 references
    public string maSanh { get; set; }

    /*Navigation Properties*/
    [ForeignKey("maSanh")]
    0 references
    public SanhModel sanh { get; set; }

    0 references
    public ICollection<BookPhongModel> BookPhongs { get; set; }
}
```

TÀI LIỆU THỰC HÀNH MODEL ASP.NET CORE

- Sử dụng `Display` (`Name = "Tên thay thế"`) để sử dụng cho việc xuất ra View
 - Thêm: `using System.ComponentModel.DataAnnotations;`
 - Tại mỗi thuộc tính ta thêm như sau:

```
21 references
public class PhongModel
{
    [Display(Name = "Mã Phòng")]
    15 references
    public string maPhong { get; set; }

    [Display(Name = "Tên Phòng")]
    12 references
    public string tenPhong { get; set; }

    [Display(Name = "Sức Chứa")]
    12 references
    public int sucChua { get; set; }

    [Display(Name = "Loại Phòng")]
    12 references
    public string loaiPhong { get; set; }
}
```

- Giới hạn kiểu dữ liệu và in thông báo lỗi khi vi phạm phạm vi dữ liệu
 - Ví dụ giới hạn tên phòng kiểu chuỗi 40 ký tự `varchar(40)`
 - Giới hạn trong ASP.NET core là `string(40)` như sau:

```
[Display(Name = "Tên Phòng")]
[StringLength(40, ErrorMessage = "Tên phòng phải < 40 ký tự")]
12 references
public string tenPhong { get; set; }
```

- Format Datetime theo dạng dd-mm-yyyy

```
[DisplayFormat(DataFormatString = "{0:dd-MM-yyyy}", ApplyFormatInEditMode = true)]
12 references
public DateTime ngaySinh { get; set; }
```

- Có thể ghép thuộc tính như sau:

```
[Display(Name = "Full Name")]
public string FullName
{
    get
    {
        return LastName + ", " + FirstMidName;
    }
}
```

TÀI LIỆU THỰC HÀNH MODEL ASP.NET CORE

- Giả sử thuộc tính trong model là TenKH mà trong CSDL lưu là TenK để mapping với thuộc tính trong CSDL ta dùng DisplayAttribute như sau:

```
[Column("TenK")]
3 references
public string tenKH { get; set; }
```

- Có thể cài đặt khóa chính của Model mình bằng cách thêm DisplayAttribute như sau:

```
[Key]
12 references
public string maKH { get; set; }
```

- **Câu 1:** Sinh viên cài các Model còn lại trong lược đồ cơ sở dữ liệu “quản lý nhà hàng”
- Tạo Model với ID tự động tăng:
 - Thêm DDL sau vào CSDL Quản lý nhà hàng

```
CREATE TABLE NHANVIEN (
    MANV INT IDENTITY(1,1) PRIMARY KEY,
    TENNV NVARCHAR(50)
)
```

- Tạo Model nhân viên như sau:

```
19 references
public class NhanVienModel
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    11 references
    public int manv { get; set; }
    12 references
    public string tennv { get; set; }
}
```

[DatabaseGenerated(DatabaseGeneratedOption.Identity)]:

Dùng để tạo tự động trong database tăng dần mã id



TÀI LIỆU THỰC HÀNH MODEL ASP.NET CORE

➤ **Câu 2:** Sinh viên cài các Model còn lại có id/khóa chính tự động tăng trong lược đồ cơ sở dữ liệu “quản lý nhà hàng”.

- **Bước 2:** Tạo kết nối cơ sở dữ liệu trong file “appsetting.json”

```
"ConnectionStrings": {  
  "DefaultConnection": "Server=localhost; Database=qlnh;  
Trusted_Connection=True; MultipleActiveResultSets=true"  
},
```

- **Bước 3:** Tạo database Context.
 - Trong Project ASP.NET core tạo một thư mục tên là: **DATA**
 - Trong thư mục DATA tạo một tập tin **RestaurantContext.cs** có cấu trúc như sau:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Threading.Tasks;  
using Microsoft.EntityFrameworkCore;  
using ProjectTraining.Models;  
  
namespace ProjectTraining.Data  
{  
    public class RestaurantContext:DbContext  
    {  
        public RestaurantContext(DbContextOptions<RestaurantContext>  
options) : base(options)  
        {  
  
            public DbSet<KhachHangModel> KhachHangs { get; set; }  
            public DbSet<TiecModel> Tiecs { get; set; }  
            public DbSet<BookPhongModel> BookPhongs { get; set; }  
            public DbSet<PhongModel> Phongs { get; set; }  
            public DbSet<SanhModel> Sanhs { get; set; }  
            public DbSet<NhanVienModel> NhanViens { get; set; }  
  
            protected override void OnModelCreating(ModelBuilder  
modelBuilder)  
            {  
                modelBuilder.Entity<KhachHangModel>().ToTable("khachhang").  
                    HasKey(c=> c.maKH );  
            }  
        }  
    }  
}
```

```
modelBuilder.Entity<TiecModel>().ToTable("tiec").
    HasKey(c=>c.maTiec);

modelBuilder.Entity<BookPhongModel>().ToTable("bookphong").
    HasKey(c => new { c.maTiec, c.maPhong }); ;

modelBuilder.Entity<PhongModel>().ToTable("phong").
    HasKey(c=>c.maPhong);

modelBuilder.Entity<SanhModel>().ToTable("sanh").
    HasKey(c=>c.maSanh);

modelBuilder.Entity<NhanVienModel>().ToTable("nhanvien");
    }
}
```

- **Bước 4:** Vào file Startup.cs để chỉnh sửa **ConfigureService** như sau:

```
// This method gets called by the runtime. Use this method to add services to the container.
0 references
public void ConfigureServices(IServiceCollection services)
{
    services.AddControllersWithViews();

    services.AddDbContext<RestaurantContext>{options =>
        options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection"))};
}
```

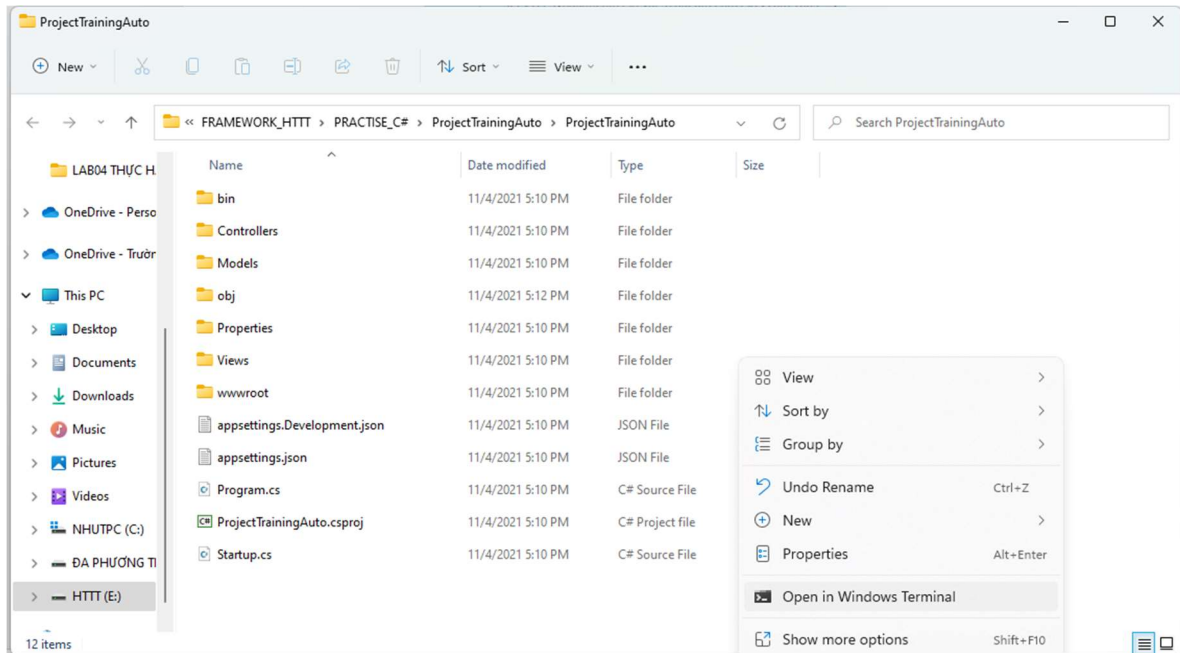
C. THỰC HIỆN TẠO MODEL BẰNG CÔNG CỤ SCAFFOLDED

- **Bước 1:** Chọn Project ASP.NET Core → Chọn **Open Folder in File Explore**



TÀI LIỆU THỰC HÀNH MODEL ASP.NET CORE

- **Bước 2:** Tại thư mục vừa mở, mở cmd với quyền admin



Gõ lệnh như sau:

```
dotnet ef dbcontext scaffold  
"Server=localhost;Database=qlnh;Trusted_Connection=True;"  
Microsoft.EntityFrameworkCore.SqlServer --output-dir Models
```

- Kiểm tra lại kết quả auto scaffold bằng thư mục Model.