



# **LẬP TRÌNH C# 4**

## **BÀI 7: ENTITY FRAMEWORK CORE (P1)**

- ⊙ Giới thiệu EF core
- ⊙ EF Core cho Asp core project
- ⊙ Tạo web với EF core database first



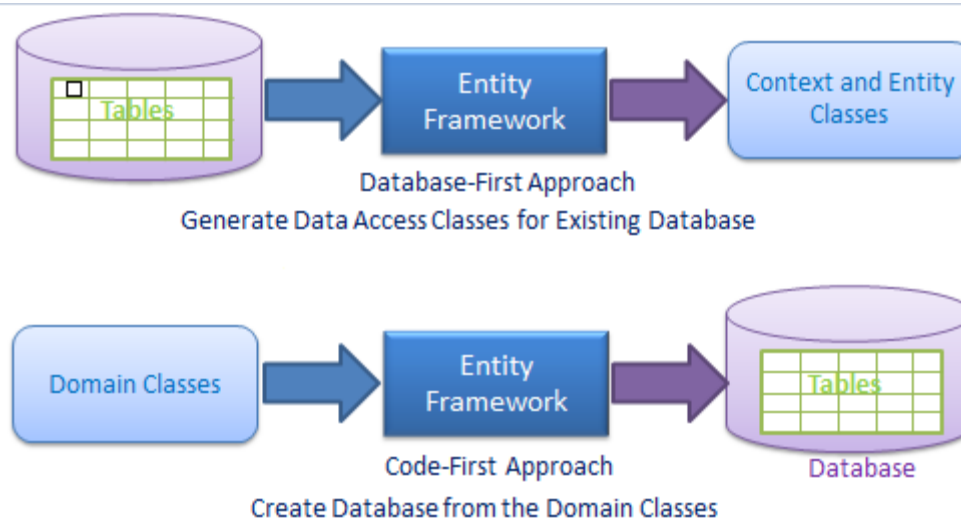
- ❑ Entity Framework Core là phiên bản mới của Entity Framework sau EF 6.x (**Xem lại Entity Framework 6.x tại slide 5 môn C#3**)
- ❑ Mã nguồn mở, nhẹ, có thể mở rộng và là phiên bản đa nền tảng của công nghệ truy cập dữ liệu Entity Framework.
- ❑ Là một khung đối tượng / Ánh xạ quan hệ (O / RM). Đây là một cải tiến của ADO.NET, cung cấp cho các nhà phát triển một cơ chế tự động để truy cập và lưu trữ dữ liệu trong cơ sở dữ liệu.
- ❑ EF giúp lập trình viên giảm thiểu việc lập trình mã nguồn cần thiết để truy cập và tương tác với cơ sở dữ liệu

## □ Các ứng dụng Entity Framework Core hỗ trợ:

Application Types	<u>ASP.NET Core Applications</u> Web, API, Console, etc.	<u>.NET 4.5+ Applications</u> Console, WinForm, WPF, ASP.NET	Devices + IoT, Mobile, PC, Xbox, Surface Hub	<u>Mobile Application</u> Android, iOS, Windows
EF Core	EF Core	EF Core	EF Core	EF Core
Framework	.NET Core	.NET 4.5+	UWP	Xamarin
OS	Windows, Mac, Linux	Windows	Windows 10	Mobile

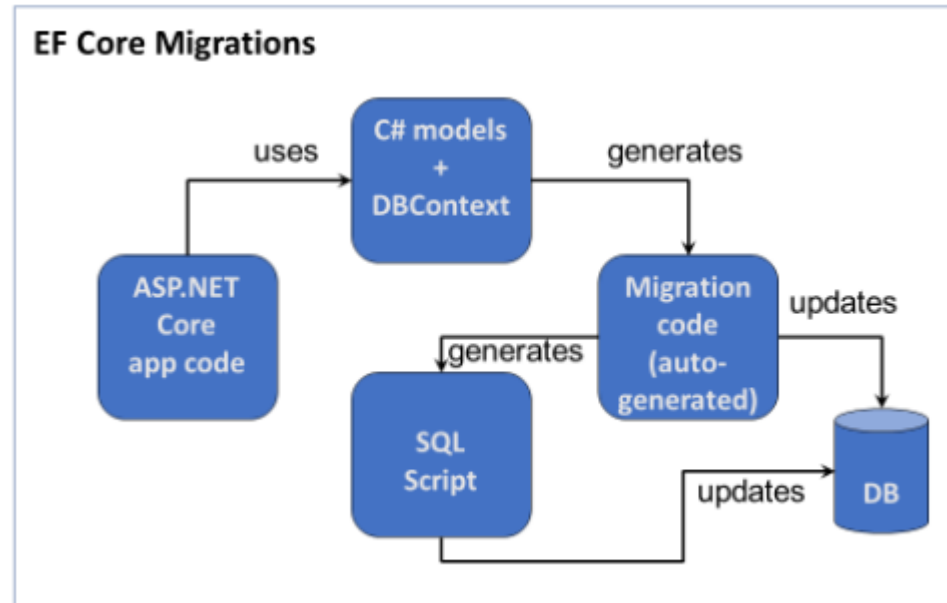
## □ Tham khảo so sánh **EF Core & EF6**: <https://docs.microsoft.com/en-us/ef/efcore-and-ef6/>

- Entity Framework Core hỗ trợ 2 hướng tiếp cận Code-First và Database-First



- Code-First: EF Core API tạo database dựa trên các lớp mô hình đối tượng thiết kế trước
- Database-First: EF Core API tạo các lớp mô hình đối tượng dựa trên database thiết kế trước

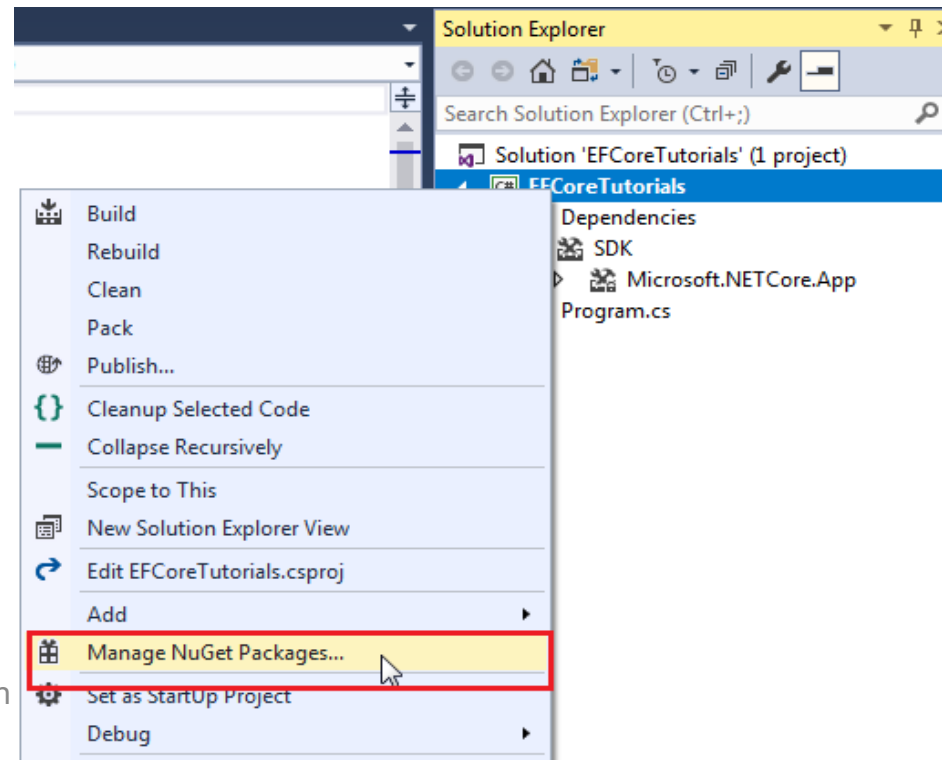
- ❑ Sử dụng EF Core Migrations có thể thêm và thao tác các đối tượng cơ sở dữ liệu, xác định mối quan hệ giữa các thực thể, đồng bộ các phiên bản csdl



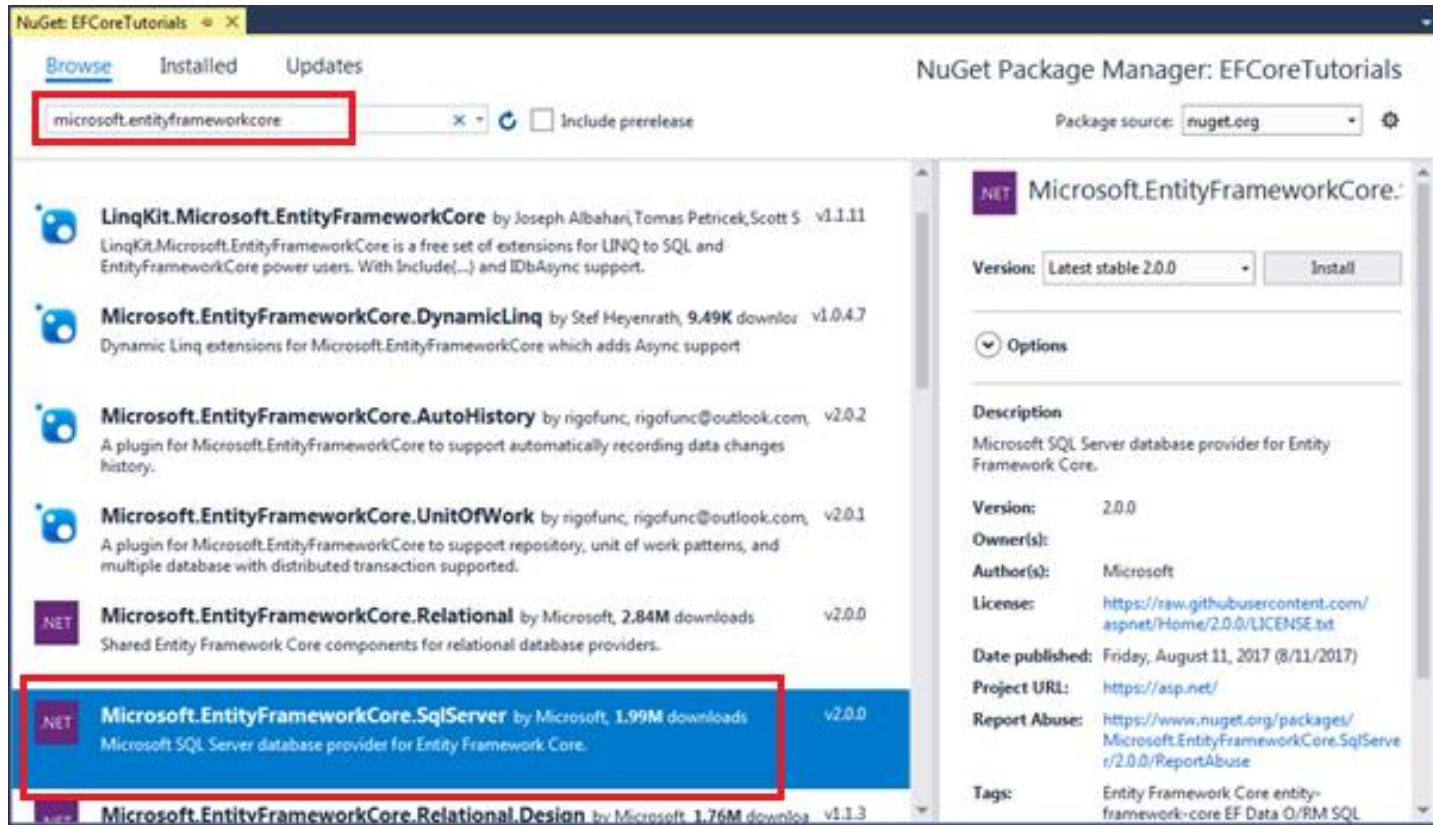
- ❑ Một số lệnh Migrations thông dụng: Add-Migration <migration-name>, Update-Database...

❑ EF Core tương tác với nhiều loại database thông qua các provider → cài EF Core DB Provider, giả sử tương tác với MS SQL Server database → cài Microsoft.EntityFrameworkCore.SqlServer, có 2 cách cài:

❖ Dùng NuGet Packages:

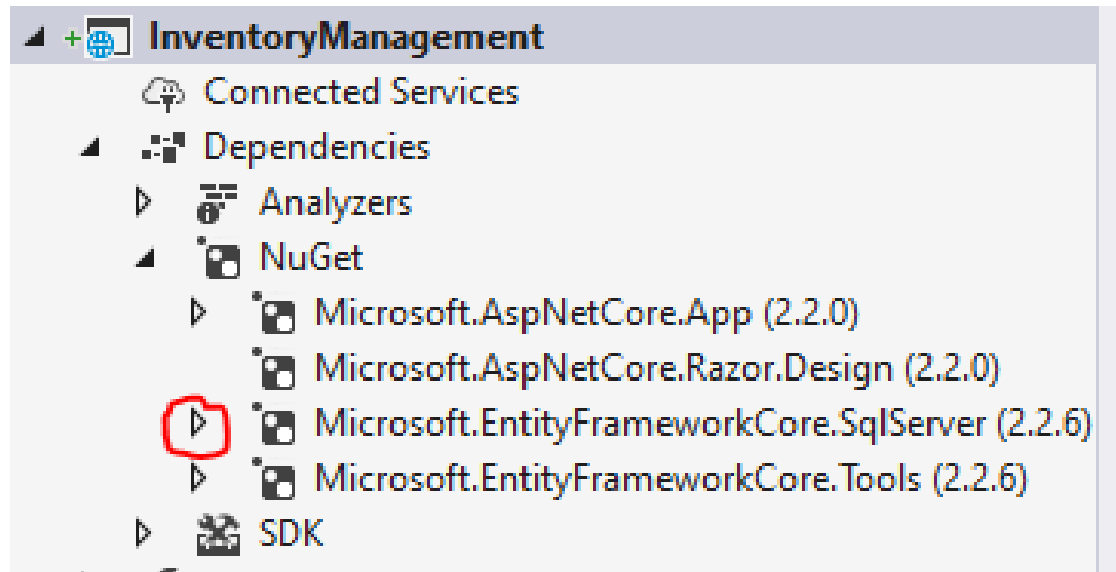


## ❖ Dùng NuGet Packages:





## ❖ Dùng NuGet Packages:

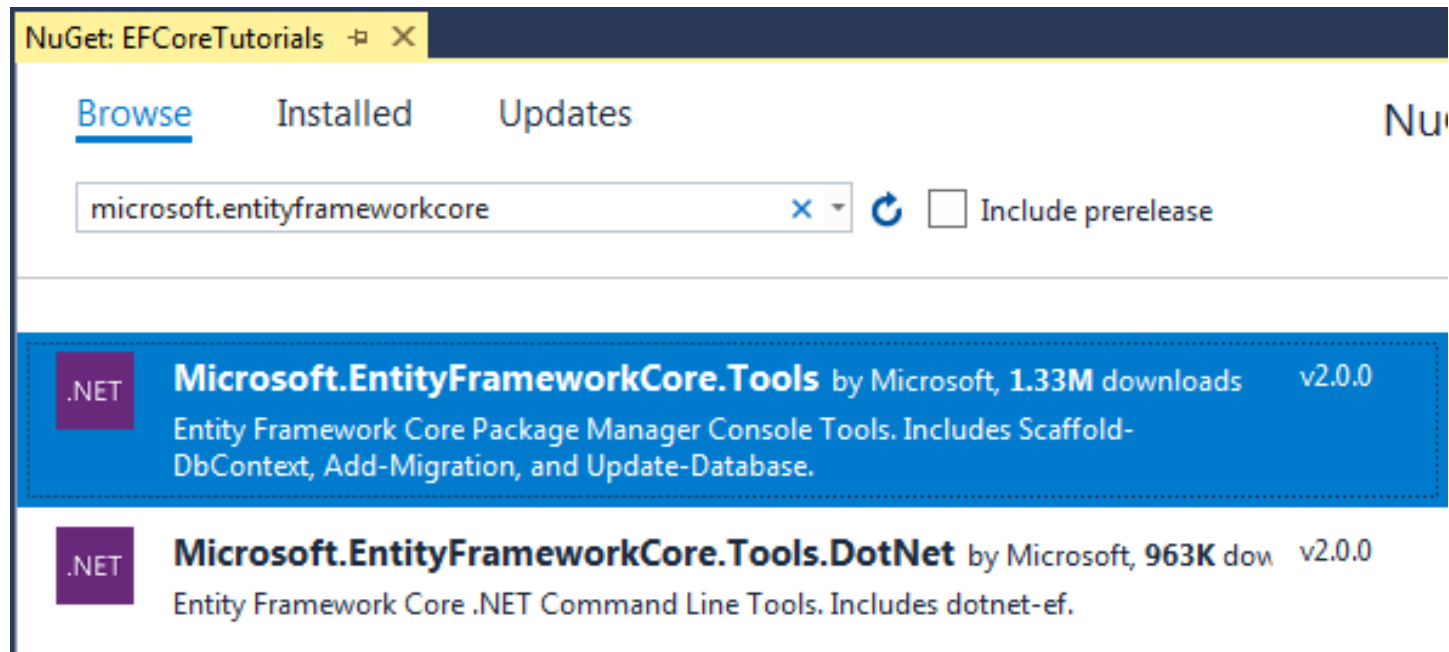


## ❑ Dùng Package Manager Console: Tools -> NuGet Package Manager -> Package Manager Console

```
PM> Install-Package Microsoft.EntityFrameworkCore.SqlServer
```

❑ EF Core hỗ trợ mạnh giao diện dòng lệnh: migrations, tạo entity từ database...=> Install EF Core Tools cho:

❖ Package Manager Console (PMC)



- ```
a:\> cd "C:\Program Files\Microsoft\dotnet\tools\dotnet-ef"
a:\Program Files\Microsoft\dotnet\tools\dotnet-ef> dotnet tool install --global dotnet-ef
You can invoke the tool using the following command: dotnet-ef
Tool 'dotnet-ef' (version '3.1.4') was successfully installed.
```

```

Administrator: Command Prompt
a:\fpol\SP2017\C#4\Slide\slide7\InventoryManagement>dotnet ef

Entity Framework Core .NET Command-line Tools 3.1.4

Usage: dotnet ef [options] [command]

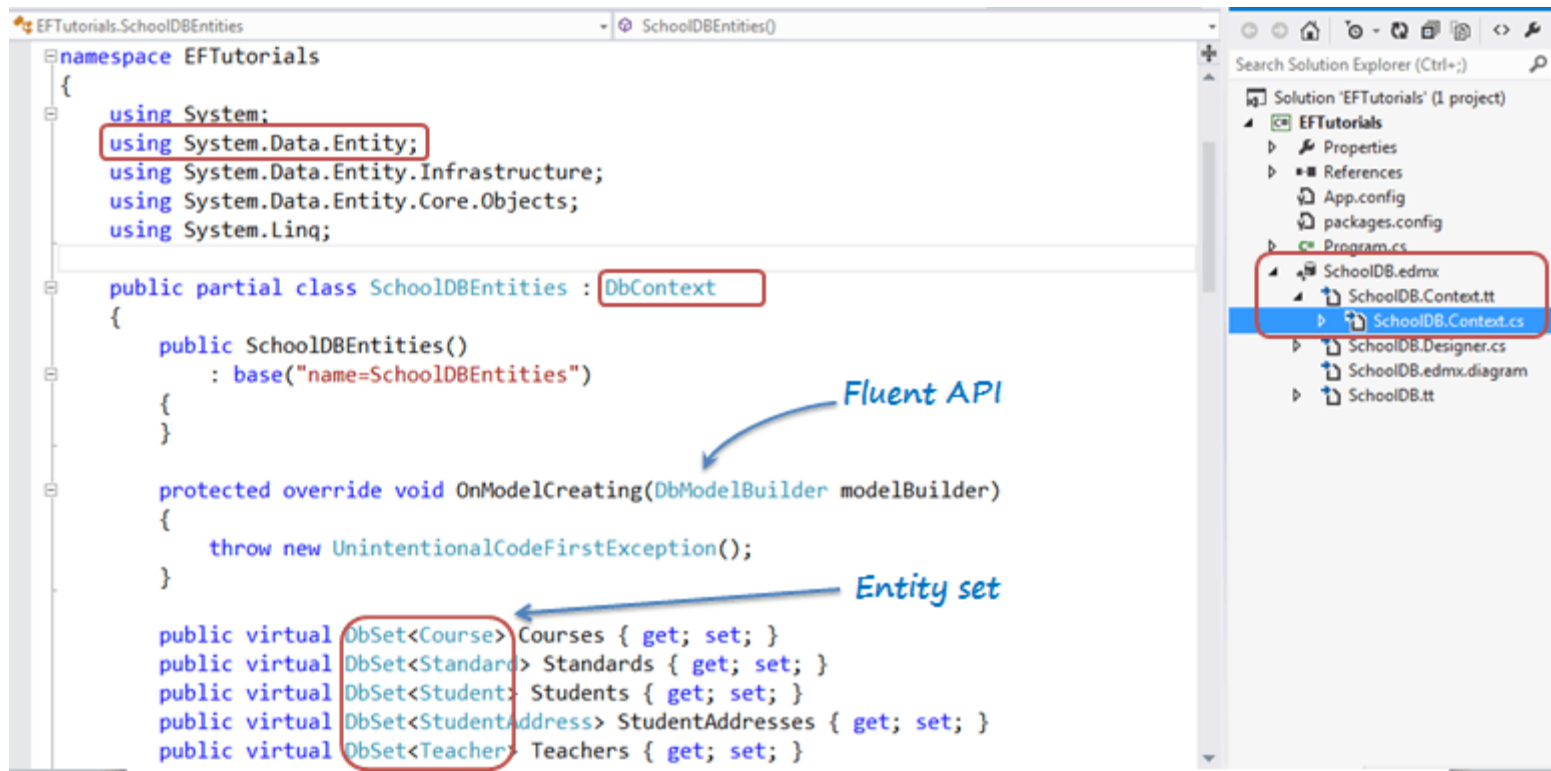
Options:
  --version             Show version information
  -h|--help             Show help information
  -v|--verbose          Show verbose output.
  --no-color            Don't colorize output.
  --prefix-output       Prefix output with level.

Commands:
  database              Commands to manage the database.
  dbcontext             Commands to manage DbContext types.
  migrations            Commands to manage migrations.

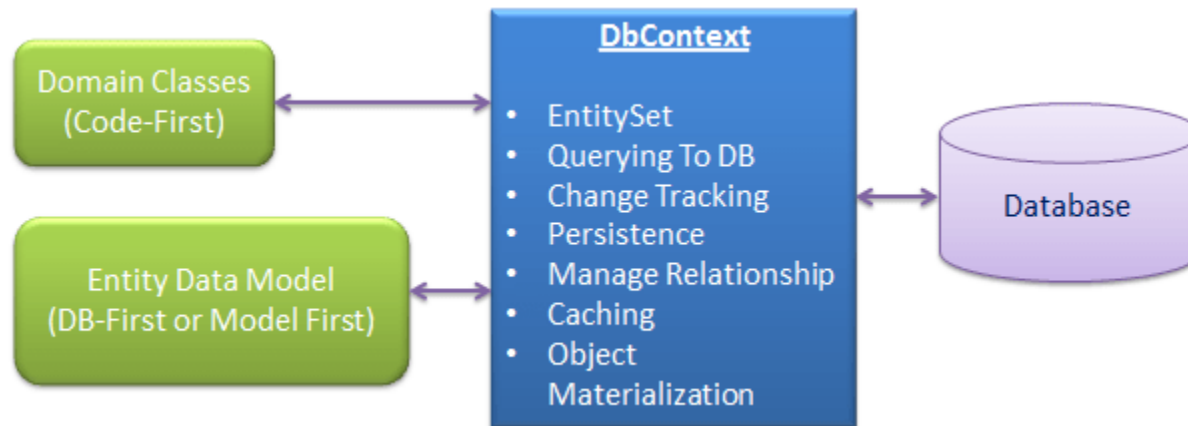
Use "dotnet ef [command] --help" for more information about a command.

```

- ❑ EF core Entity có thể query, insert, update ,delete dữ liệu thông qua DbContext
- ❑ Là một cầu nối giữa lớp domain hoặc thực thể (Entities)và CSDL



- ❑ DbContext là lớp chính chịu trách nhiệm cho việc tương tác với dữ liệu như là đối tượng. DbContext chịu trách nhiệm cho các hoạt động sau:



- ❑ **EntitySet:** DbContext chứa tập thực thể (DbSet<TEntity>) cho tất cả thực thể nối với những bảng của CSDL
- ❑ **Querying:** DbContext chuyển đổi những truy vấn LINQ-to-Entities thành truy vấn SQL và gửi nó tới CSDL.
- ❑ **Change Tracking:** Theo dõi những thay đổi xảy ra trong những thực thể sau khi nó đã truy vấn từ CSDL
- ❑ **Persisting Data:** Thực hiện các thao tác Insert, Update và Delete tới CSDL dựa trên những gì mà thực thể thể hiện
- ❑ **Caching:** DbContext mặc định thực hiện caching mức đầu tiên. Nó lưu những thực thể đã được nhận suốt vòng đời của một lớp context
- ❑ **Manage Relationship:** DbContext cũng quản lý những quan hệ sử dụng CSDL, MSL và SSDL trong phương pháp tiếp cận DB-First hoặc Model-First hoặc sử dụng fluent API trong phương pháp tiếp cận Code-First
- ❑ **Object Materialization:** DbContext chuyển đổi bảng dữ liệu thô vào những đối tượng thực thể.

# CÁC PHƯƠNG THỨC – THUỘC TÍNH DbContext

| Phương thức      | Nhiệm vụ                                                                                                                              |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Entry            | Lấy thông tin của entity để truy xuất các thông tin liên quan đến sự thay đổi của entity, hoặc thực hiện các thao tác đối với entity. |
| SaveChanges      | Tạo và thực thi các truy vấn INSERT, UPDATE và DELETE đối với các entity có trạng thái tương ứng (Added, Modified, Deleted).          |
| SaveChangesAsync | Phiên bản bất đồng bộ của SaveChanges()                                                                                               |
| Set              | Tạo object của DbSet<Tentity> dùng để truy vấn và lưu các biến của Tentity.                                                           |
| OnModelCreating  | Phương thức virtual cho phép bạn ghi đè trong lớp context để thực hiện cấu hình cho cơ sở dữ liệu sử dụng fluent API.                 |

| Property      | Nhiệm vụ                                                                                                                                           |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| ChangeTracker | Property này cho phép truy xuất tới Change Tracker API, tức là truy xuất tới thông tin và hoạt động của các entity đang được context này theo dõi. |
| Configuration | Truy xuất tới các thông tin cấu hình.                                                                                                              |
| Database      | Truy xuất tới các thông tin và hoạt động của cơ sở dữ liệu.                                                                                        |

- ❑ Là tập hợp các entity object hỗ trợ thực hiện các lệnh tạo mới/ đọc/ cập nhật/ xóa các entity

| Phương thức          | Nhiệm vụ                                                                                                                                                                                                                              |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add                  | Thêm entity object mới vào danh sách.                                                                                                                                                                                                 |
| AsNoTracking<Entity> | Đây là phương thức kế thừa từ DbQuery, trả về một danh sách entity nhưng các entity trong đó không được cached (và cũng không được track) bởi context. Sử dụng phương thức này nếu cần các entity chỉ đọc để tăng hiệu suất truy vấn. |
| Attach(Entity)       | Gắn (attach) một entity đã có vào tập hợp. Nhiệm vụ của Attach đơn giản là báo cho Entity Framework biết rằng đây là một entity đã có sẵn, thay vì phải tải nó từ CSDL.                                                               |
| Create               | Tạo một entity mới nhưng không được thêm vào danh sách quản lý của DbSet.                                                                                                                                                             |
| Find(int)            | Tìm một entity đang được theo dõi trong context.                                                                                                                                                                                      |
| Include              | Yêu cầu tải thêm các entity có quan hệ (1-n, n-n, 1-1).                                                                                                                                                                               |
| Remove               | Đánh dấu xóa cho entity.                                                                                                                                                                                                              |
| SqlQuery             | Chạy truy vấn SQL.                                                                                                                                                                                                                    |

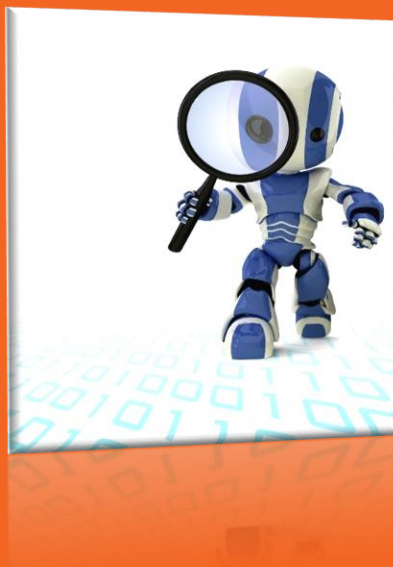




# DEMO

- Tạo ASP.NET Core Web Application
- Cài EF Core
- EF Core Tools





# **LẬP TRÌNH C# 4**

## **BÀI 7: ENTITY FRAMEWORK CORE (P2)**

## ❑ Các bước thực hiện:

1. Prerequisites
2. Database creation
3. Create an ASP.NET Core application
4. Install necessary NuGet packages
5. Connect application to database
6. Scaffold CRUD operations
7. Run application

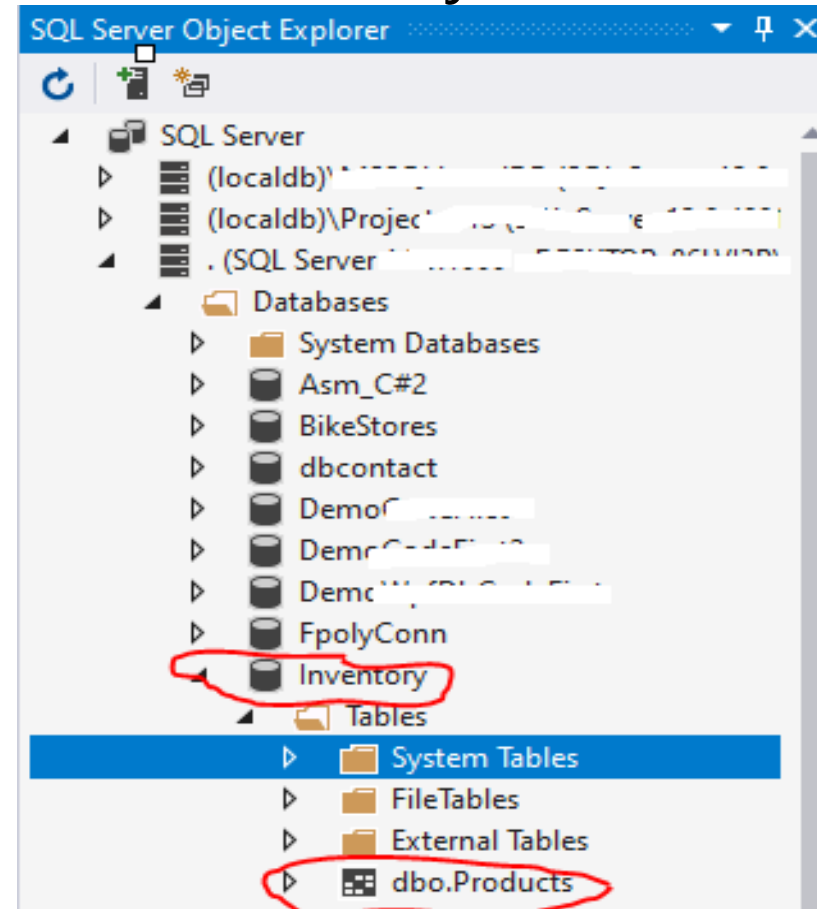
## ❑ Bước1 Prerequisites:

- ❖ Công cụ Visual Studio 2019 16.3.0 (sv chọn version phù hợp) - <https://visualstudio.microsoft.com/downloads/>
- ❖ Cài NET Core SDK 2.2 (sv chọn version phù hợp) - <https://dotnet.microsoft.com/download>
- ❖ Sql server 2016 (sv chọn version phù hợp) - <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

## ❑ Bước2 Tạo csdl:

- ❖ Sử dụng Sql Server Express hoặc SQL Server Object Explorer tích hợp trong Vs2019
- ❖ Tạo database tên "Inventory"
- ❖ Tạo table Products:

```
Create Table Products(  
    ProductId BigInt Identity(1,1) Primary Key,  
    Name Varchar(100) Not Null,  
    Category Varchar(100),  
    Color Varchar(20),  
    UnitPrice Decimal Not Null,  
    AvailableQuantity BigInt Not Null,  
    CratedDate DateTime Default(GetDate()) Not null)
```





## ❑ Bước 3 tạo ứng dụng ASP.NET Core (Model – View-Controller)


### Create a new ASP.NET Core web application


.NET Core


ASP.NET Core 3.0


**Empty**  
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

**API**  
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

**Worker Service**  
An empty project template for creating a worker service.

**Web Application**  
A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content.

**Web Application (Model-View-Controller)**  
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

**gRPC** **gRPC Service**

**Authentication**  
No Authentication  
[Change](#)

**Advanced**  
☐ Configure for HTTPS  
☐ Enable Docker Support  
(Requires [Docker Desktop](#))  

Linux

Author: Microsoft  
Source: SDK 3.0.100-preview7-012821

[Get additional project templates](#)

Back

Create

## ❑ Bước 3 tạo ứng dụng ASP.NET Core (Model – View–Controller)

Create a new ASP.NET Core web application

.NET Core ASP.NET Core 3.0

**Empty**  
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

**API**  
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

**Worker Service**  
An empty project template for creating a worker service.

**Web Application**  
A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content.

**Web Application (Model-View-Controller)**  
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

gRPC gRPC Service

Solution 'InventoryManagement' (1 of 1 project)

**InventoryManagement**

- Connected Services
- Dependencies
- Properties
- wwwroot
- Controllers
- Models
- Views
- appsettings.json
- Program.cs
- Startup.cs

Authentication: No Authentication  
[Change](#)

**Advanced**

☐ Configure for HTTPS

☐ Enable Docker Support  
(Requires [Docker Desktop](#))

Linux

Author: Microsoft  
Source: SDK 3.0.100-preview7-012821

[Back](#) [Create](#)

- ❑ Bước 4 cài các packets hỗ trợ EF (tham khảo “Cài đặt EF Core cho Asp core project”-slide7)
  - ❖ Package `Microsoft.VisualStudio.Web.CodeGeneration.Design`: hỗ trợ phát sinh controller và view
  - ❖ Package `Microsoft.EntityFrameworkCore.Tools`: hỗ trợ tạo context class và model class từ database
  - ❖ Package `Microsoft.EntityFrameworkCore.SqlServer` : hỗ trợ provider tương tác sql server



- ❑ Bước 5 tạo mối kết hợp giữa ứng dụng và csdl bằng tính năng Scaffolding của EF core giúp tự động:
  - ❖ Tạo các entity, DbContext class từ csdl
  - ❖ Tạo các module create, read, update, and delete (CRUD) tương ứng trong Models, Views, Controller
- ❑ Kích hoạt tính năng Scaffolding của EF core bằng Nuget PMC:

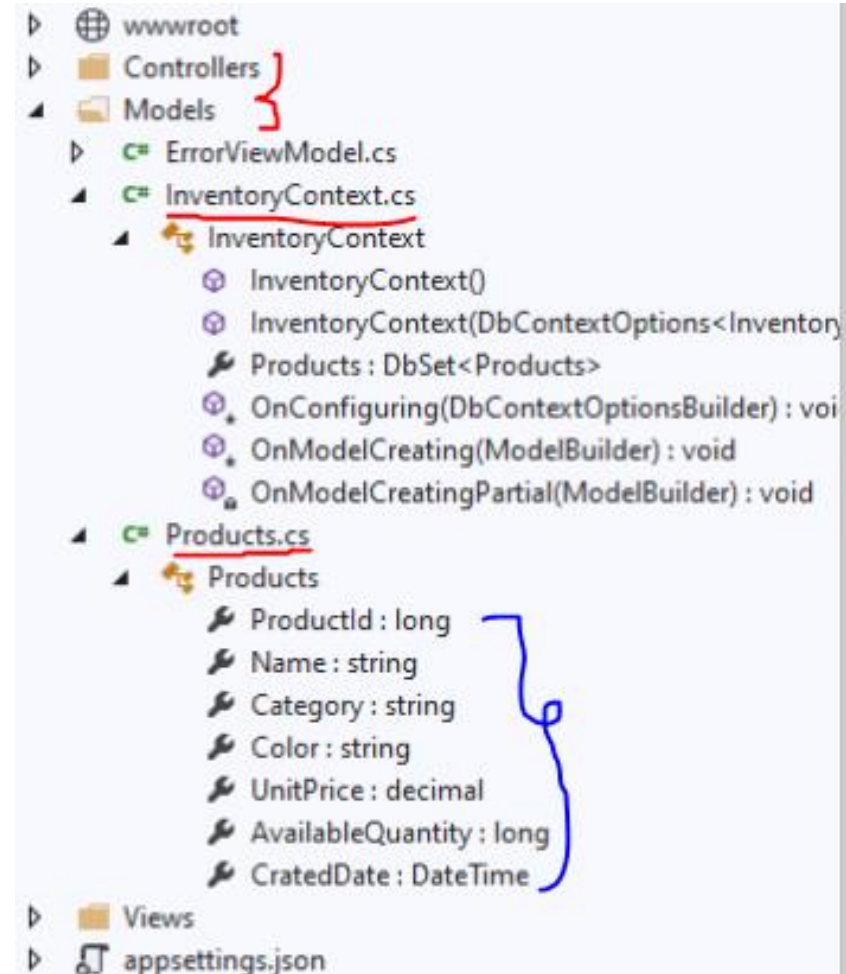
```
Scaffold-DbContext "Server=ABCSERVER;Database=Inventory;Integrated  
Security=True" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
```

- ❖ Server: server name của sql server
- ❖ Database: tên csdl
- ❖ OutputDir: thư mục chứa các entity và context class được sinh ra từ csdl

❑ Bước 5 tạo thành công thì sẽ có các

Thư mục Models, Controller  
Và View được sinh ra.

Bên trong Models có lớp  
InventoryContext chứa các  
entity và Products.cs chứa các  
Field tương ứng các column  
Trong table của csdl



- ❑ Bước 5 tạo thành công, bên trong InventoryContext.cs có phương thức OnConfiguring cấu hình cứng chuỗi connecstring kết nối với csdl

0 references

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    if (!optionsBuilder.IsConfigured)
    {
        #warning To protect potentially sensitive information in your connection string, you should
        optionsBuilder.UseSqlServer("Server=*****;Database=Inventory;Trusted_Connection=True;");
    }
}
```

- ❑ Bước 5 tạo thành công, có thể bỏ phương thức OnConfiguring và cấu hình động connecstring trong appsettings.json

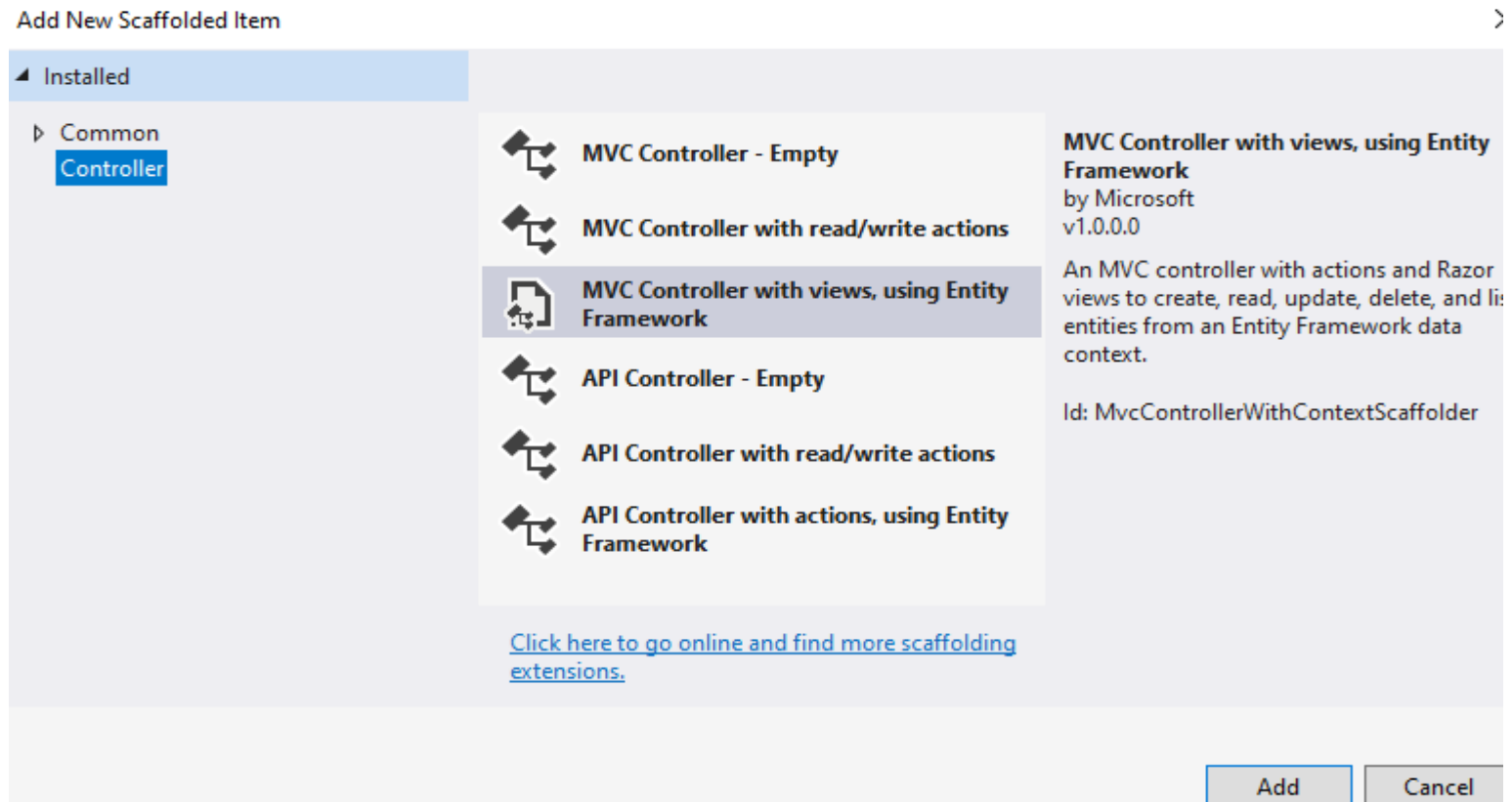
```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "InventoryDatabase": "Server=SYNCLAPN18531;Database=Inventory;Trusted_Connection=True;"
  }
}
```

- ❑ Ứng dụng sẽ đọc kết nối trong ConfigureServices

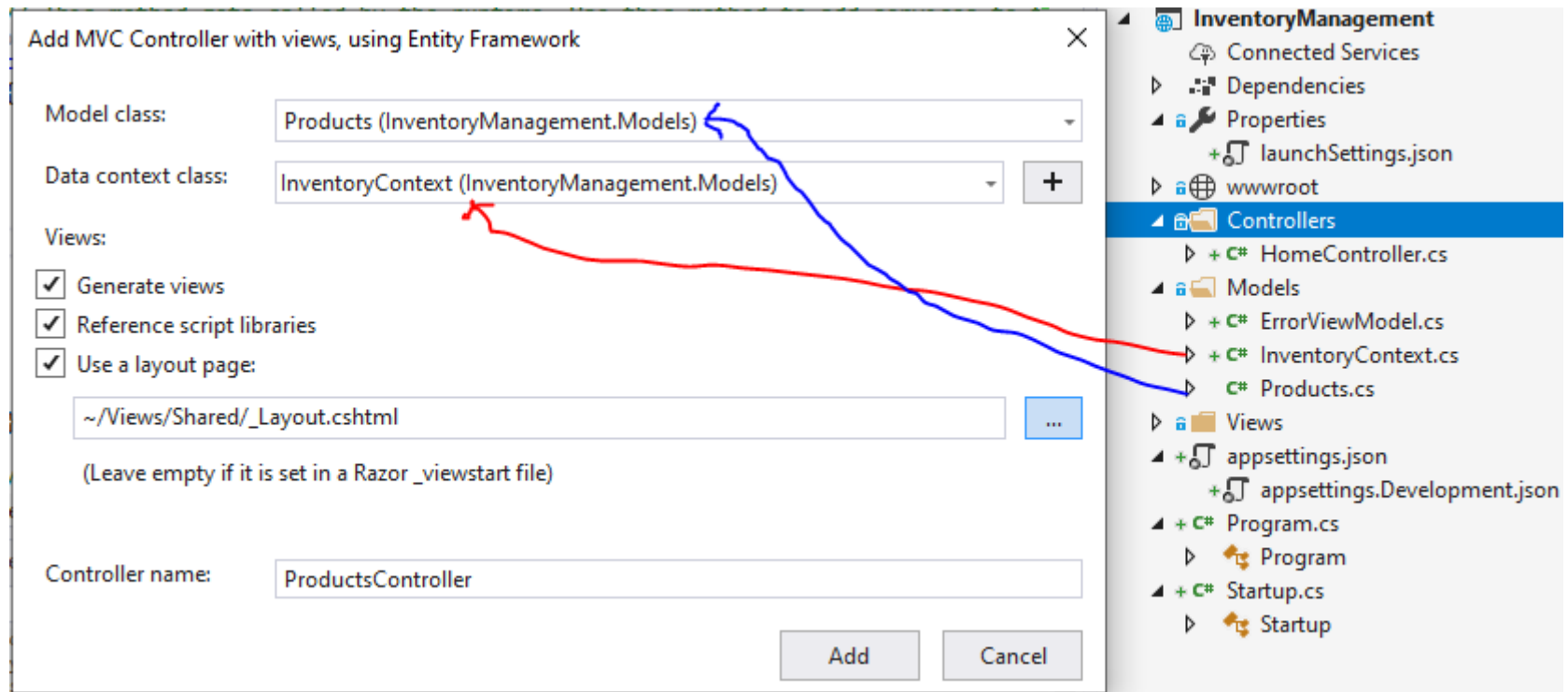
```
public void ConfigureServices(IServiceCollection services)
{
    var connection = Configuration.GetConnectionString("InventoryDatabase");
    services.AddDbContext<InventoryContext>(options => options.UseSqlServer(connection));

    services.AddControllersWithViews();
}
```

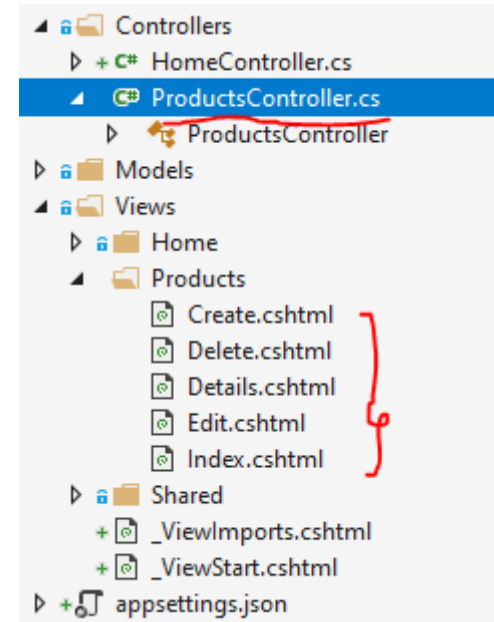
- ❑ Bước 6 hiện thực CRUD operations: Right-click trên controller , chọn **add new item**, chọn **controller**, chọn **MVC Controller with views, using Entity Framework** và chọn **Add**.



- ❑ Bước 6 hiện thực CRUD operations
- ❑ Chọn model và data context class



- ❑ Bước 6 hiện thực CRUD operation thành công sẽ tự sinh ra các action method tương ứng các thao tác CRUD và tự sinh các view tương ứng
- ❑ Bước 7 thiết lập Products controller chạy đầu tiên để kiểm tra kết quả



```
app.UseMvc(routes =>
{
    routes.MapRoute(
        name: "default",
        template: "{controller=Products}/{action=Index}/{id?}");
});
```

## ❑ Màn hình trang index

### Index

[Create New](#)

| Name      | Category | Color | UnitPrice | AvailableQuantity | CratedDate             |                                                                         |
|-----------|----------|-------|-----------|-------------------|------------------------|-------------------------------------------------------------------------|
| Fpoly hcm | Cntt     | Blue  | 20.00     | 1                 | 17/05/2020 12:23:48 PM | <a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a> |

## ❑ Màn hình Delete

### Delete

Are you sure you want to delete this?  
Products

|                          |                        |
|--------------------------|------------------------|
| <b>Name</b>              | Fpoly hcm              |
| <b>Category</b>          | Cntt                   |
| <b>Color</b>             | Blue                   |
| <b>UnitPrice</b>         | 20.00                  |
| <b>AvailableQuantity</b> | 1                      |
| <b>CratedDate</b>        | 17/05/2020 12:23:48 PM |

[Delete](#) | [Back to List](#)



## ❑ Màn hình Detail

### Details

#### Products

---

|                          |                        |
|--------------------------|------------------------|
| <b>Name</b>              | Fpoly hcm              |
| <b>Category</b>          | Cntt                   |
| <b>Color</b>             | Blue                   |
| <b>UnitPrice</b>         | 20.00                  |
| <b>AvailableQuantity</b> | 1                      |
| <b>CratedDate</b>        | 17/05/2020 12:23:48 PM |

[Edit](#) | [Back to List](#)

## ☐ Màn hình Create New

### Create

#### Products

Name

Category

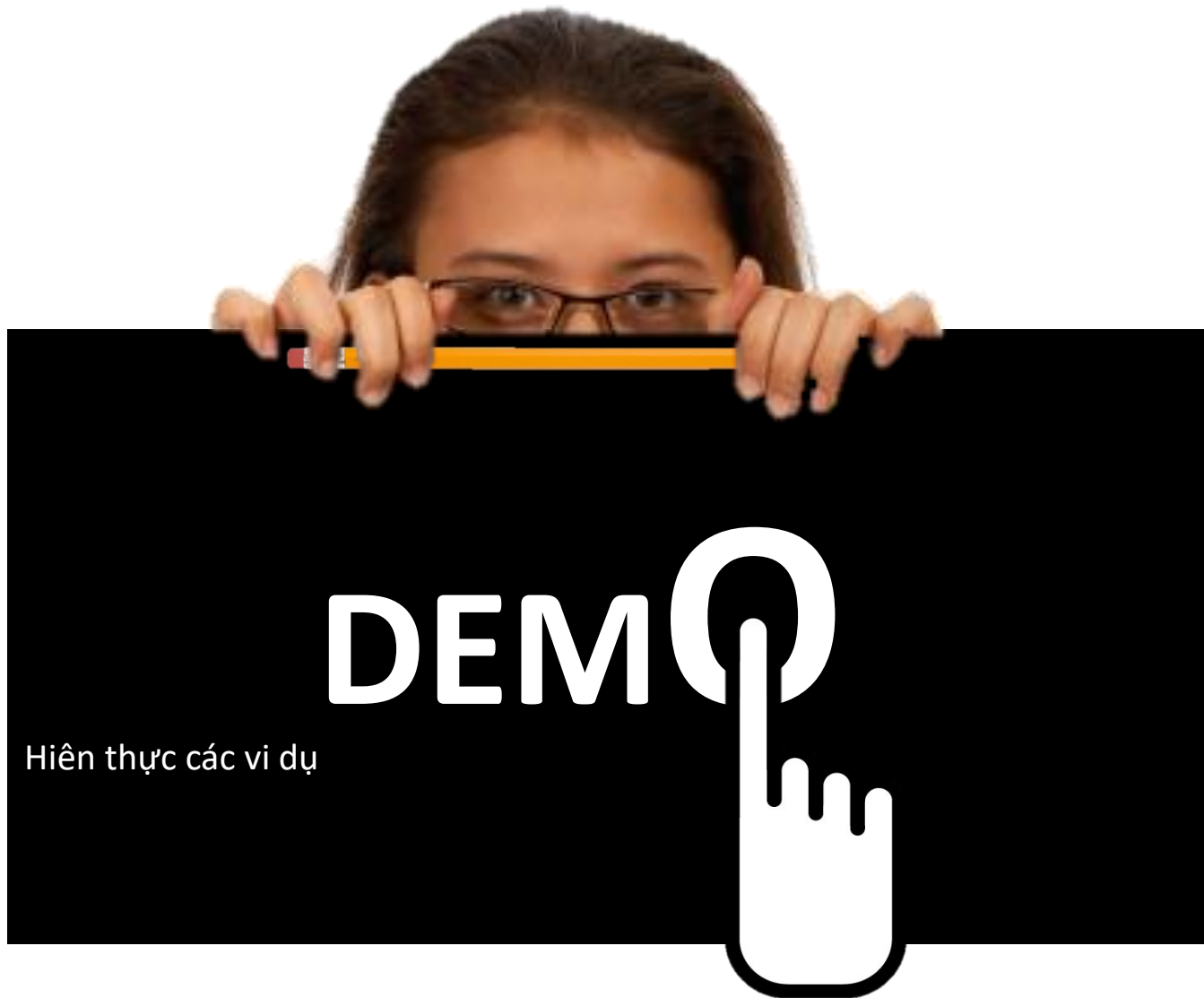
Color

UnitPrice

AvailableQuantity

Lập trình

Create



## Tổng kết bài học

- ◎Giới thiệu EF core
- ◎EF Core cho Asp core project
- ◎Tạo web với EF core database first





**KẾT THÚC**