

Bài giảng tích hợp:

ENTITY FRAMEWORK CORE

Faculty of IT

Email: smdat@hueic.edu.vn

ENTITY FRAMEWORK CORE

Chương 1. Tổng quan về Entity Framework Core

Chương 2. Truy vấn trong Entity Framework Core

Chương 3. Razor Page với Entity Framework Core

Chương 4. Asp.Net Core MVC với Entity Framework Core

Chương 5: API

Chương 1. Tổng quan về Entity Framework Core

- 1.1. Giới thiệu Entity Framework Core
- 1.2. Cài đặt Entity Framework Core
- 1.3. Tạo mô hình từ database có sẵn trong EF Core
- 1.4. DbContext trong EF Core
- 1.5. Tạo ứng dụng EF Core đầu tiên
- 1.6. Các quy ước trong EF Core

- **Entity Framework Core** là phiên bản mới của **Entity Framework** sau **EF 6.x**. Nó là mã nguồn mở, nhẹ, có thể mở rộng và là phiên bản đa nền tảng của công nghệ truy cập dữ liệu **Entity Framework**.
- **Entity Framework** là một framework **Object/Relational Mapping** (O/RM - ánh xạ quan hệ/đối tượng). 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 Core** được dự định sẽ sử dụng với các ứng dụng **.NET Core**. Tuy nhiên, nó cũng có thể được sử dụng với các ứng dụng dựa trên **.NET Framework 4.5+** tiêu chuẩn.

1.1. Giới thiệu Entity Framework Core

- Hình dưới đây minh họa các loại ứng dụng được hỗ trợ, .NET Framework và HĐH.

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

1.1. Giới thiệu Entity Framework Core

- **Lịch sử phiên bản Entity Framework Core**

Phiên bản EF Core	Ngày phát hành
Entity Framework Core 5.0	Tháng 11 - 2020 (dự kiến)
Entity Framework Core 3.1	Tháng 12 - 2019
Entity Framework Core 3.0	Tháng 9 - 2019
Entity Framework Core 2.2	Tháng 12 - 2018
Entity Framework Core 2.1	Tháng 5 - 2018
Entity Framework Core 2.0	Tháng 8 - 2017
Entity Framework Core 1.1	Tháng 11 - 2016
Entity Framework Core 1.0	Tháng 6 - 2016

1.1. Giới thiệu Entity Framework Core

- **Lịch sử phiên bản Entity Framework Core**

Mã nguồn Entity Framework Core trên GitHub:

<https://github.com/aspnet/EntityFrameworkCore>

Lộ trình phát triển Entity Framework Core:

docs.microsoft.com/en-us/ef/core/what-is-new/roadmap

Theo dõi các vấn đề của Entity Framework Core tại:

<https://github.com/aspnet/EntityFrameworkCore/issues>

Tài liệu chính thức của Entity Framework Core:

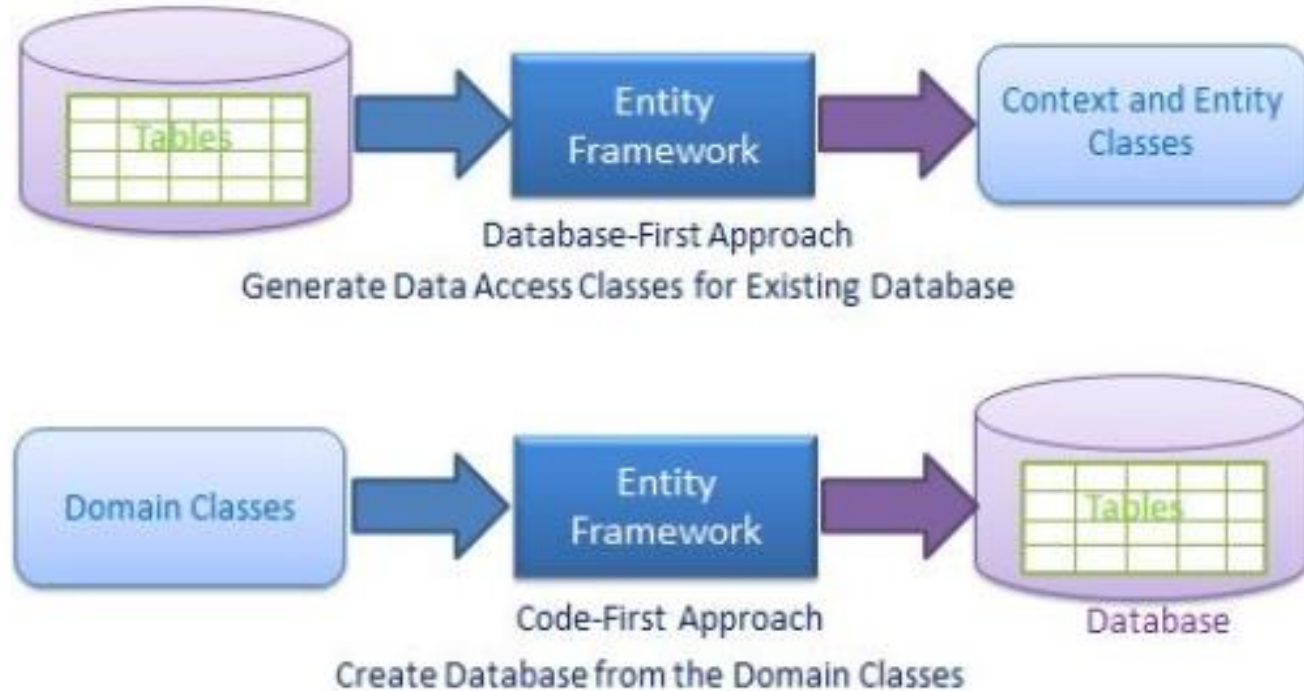
<https://docs.microsoft.com/ef/core>

- **Cách tiếp cận phát triển của EF Core**

- **EF Core** hỗ trợ hai cách tiếp cận phát triển: **Code First** và **Database First**.
- **EF Core** chủ yếu nhắm vào cách tiếp cận **Code First** và cung cấp ít hỗ trợ cho cách tiếp cận **Database First** vì trình thiết kế trực quan hoặc trình hướng dẫn cho mô hình DB không được hỗ trợ kể từ **EF Core 2.0**.
- Theo cách tiếp cận **Code First**, **EF Core API** tạo cơ sở dữ liệu và các bảng bằng cách sử dụng chuyển đổi (**migration**) dựa trên các quy ước và cấu hình được cung cấp trong các lớp thực thể của bạn. Cách tiếp cận này rất hữu ích trong thiết kế hướng miền (**Domain Driven Design - DDD**).
- Theo cách tiếp cận **Database First**, **EF Core API** tạo các lớp **thực thể** và **Context** dựa trên cơ sở dữ liệu hiện có của bạn bằng cách sử dụng các lệnh của **EF Core**. Điều này được hỗ trợ hạn chế trong **EF Core** vì nó không hỗ trợ trình thiết kế trực quan hoặc trình hướng dẫn.

1.1. Giới thiệu Entity Framework Core

- Cách tiếp cận phát triển của EF Core



1.1. Giới thiệu Entity Framework Core

- **EF Core với EF 6**
 - **Entity Framework Core** là phiên bản mới và cải tiến của **Entity Framework** cho các ứng dụng **.NET Core**. **EF Core** là phiên bản mới hoàn toàn, vì vậy nó vẫn chưa đầy đủ như **EF 6**.
 - **EF Core** tiếp tục hỗ trợ các tính năng và khái niệm giống như **EF 6** như sau:
 - DbContext và DbSet.
 - Mô hình dữ liệu.
 - Truy vấn bằng cách sử dụng Linq-to-Entities.
 - Theo dõi thay đổi.
 - SaveChanges.
 - Chuyển đổi (Migration).

- **EF Core với EF 6**

- **EF Core** dần dần sẽ có hầu hết các tính năng của **EF 6**. Tuy nhiên, có một số tính năng của **EF 6** không được hỗ trợ trong **EF Core 2.0**, chẳng hạn như:
 - EDMX/Graphical Visualization của mô hình.
 - Trình hướng dẫn mô hình dữ liệu thực thể (cho phương pháp Database First)
 - ObjectContext API.
 - Truy vấn bằng cách sử dụng Entity SQL.
 - Chuyển đổi tự động.
 - Kế thừa: Bảng trên mỗi kiểu (TPT)
 - Kế thừa: Bảng trên mỗi lớp con (TPC)
 - Mối quan hệ nhiều-nhiều.
 - Chia tách thực thể.
 - Kiểu dữ liệu không gian.
 - Ánh xạ stored procedure với DbContext cho hoạt động CUD (Create, Update, Delete).
 - Seed data.

1.1. Giới thiệu Entity Framework Core

- **EF Core với EF 6**

- **EF Core** có các tính năng mới sau đây không được hỗ trợ trong **EF 6.x**:
 - Cấu hình mối quan hệ dễ dàng
 - Các thao tác INSERT, UPDATE và DELETE hàng loạt.
 - Trình cung cấp trong bộ nhớ để thử nghiệm.
 - Hỗ trợ cho IoC (Đảo ngược sự phụ thuộc).
 - Ràng buộc duy nhất (unique constraints).
 - Thuộc tính ẩn (shadow property).
 - Khóa thay thế.
 - Bộ lọc truy vấn toàn cục.
 - Ánh xạ trường.
 - DbContext chung
 - Các mẫu tốt hơn để xử lý các biểu đồ thực thể bị ngắt kết nối

1.1. Giới thiệu Entity Framework Core

- **Trình cung cấp cơ sở dữ liệu của EF Core**
 - **Entity Framework Core** sử dụng mô hình trình cung cấp để truy cập nhiều loại cơ sở dữ liệu khác nhau. **EF Core** có các trình cung cấp dưới dạng các gói **NuGet** mà bạn cần cài đặt.
 - Bảng sau liệt kê một số trình cung cấp cơ sở dữ liệu và các gói **NuGet** cho **EF Core**.

Database	NuGet Package
SQL Server	Microsoft.EntityFrameworkCore.SqlServer
MySQL	MySql.Data.EntityFrameworkCore
PostgreSQL	Npgsql.EntityFrameworkCore.PostgreSQL
SQLite	Microsoft.EntityFrameworkCore.SQLite
SQL Compact	EntityFrameworkCore.SqlServerCompact40
In-memory	Microsoft.EntityFrameworkCore.InMemory

1.1. Giới thiệu Entity Framework Core

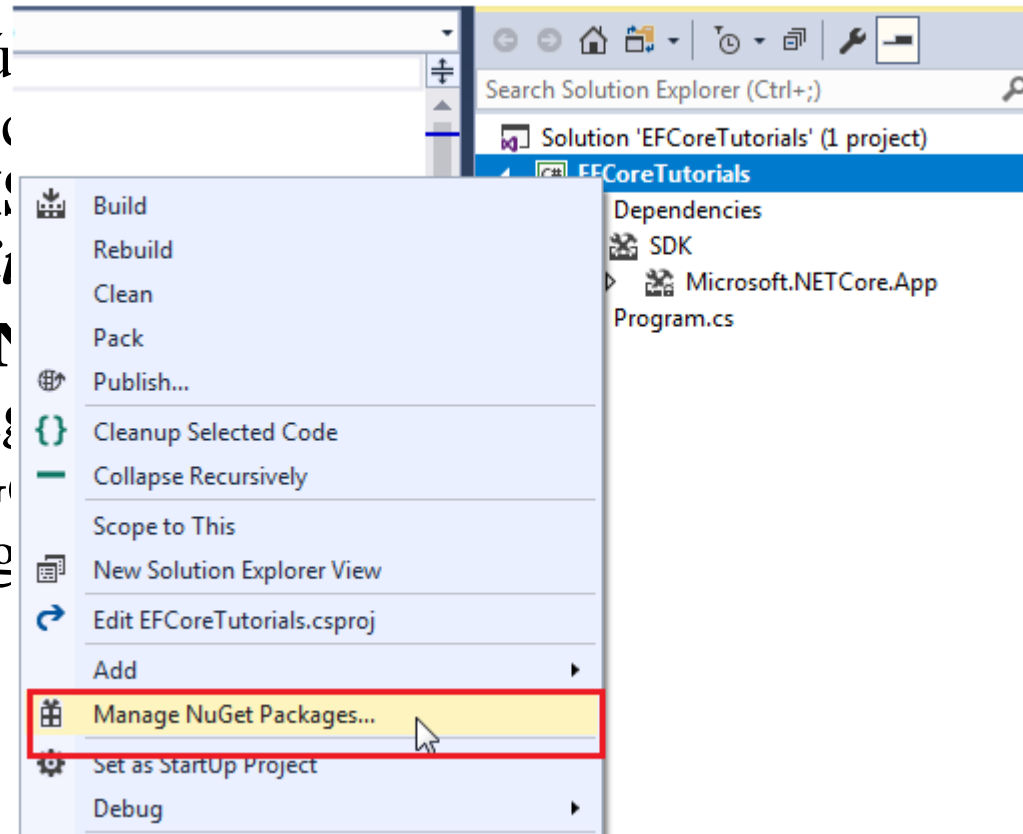
1.2. Cài đặt Entity Framework Core

- **Entity Framework Core** có thể được sử dụng với các ứng dụng dựa trên **.NET Core** hoặc **.NET Framework 4.6**. Ở phần này, sẽ tìm hiểu cách cài đặt và sử dụng **Entity Framework Core 2.0** trong các ứng dụng **.NET Core** bằng **Visual Studio 2017**.
- **EF Core** không phải là một phần của **.NET Core** và **.NET Framework** tiêu chuẩn. Nó là một gói **NuGet**. Cần cài đặt các gói **NuGet** cho hai thành phần sau đây để sử dụng **EF Core** trong ứng dụng của bạn:
 - Trình cung cấp cơ sở dữ liệu của EF Core.
 - Công cụ EF Core.

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

- **Cài đặt trình cung cấp cơ sở dữ liệu của EF Core**
 - **EF Core** cho phép chúng ta truy cập cơ sở dữ liệu thông qua mô hình trình cung cấp.
 - Có các trình cung cấp cơ sở dữ liệu **EF Core** cho các cơ sở dữ liệu khác nhau. Các trình cung cấp này có sẵn dưới dạng các gói **NuGet**.
 - Trước tiên, chúng ta cần phải cài đặt trình cung cấp cơ sở dữ liệu mà chúng ta muốn sử dụng. Trong trường hợp này, chúng ta sẽ sử dụng trình cung cấp cơ sở dữ liệu **Microsoft.EntityFrameworkCore.SqlServer**.
 - Để cài đặt gói NuGet này, chúng ta cần phải mở Visual Studio và vào dự án trong Solution Explorer, chọn **Manage NuGet Packages** (hoặc **Tools -> NuGet Package Manager -> Manage NuGet Packages for Solution**).



cấp cơ
cấp cơ
NuGet

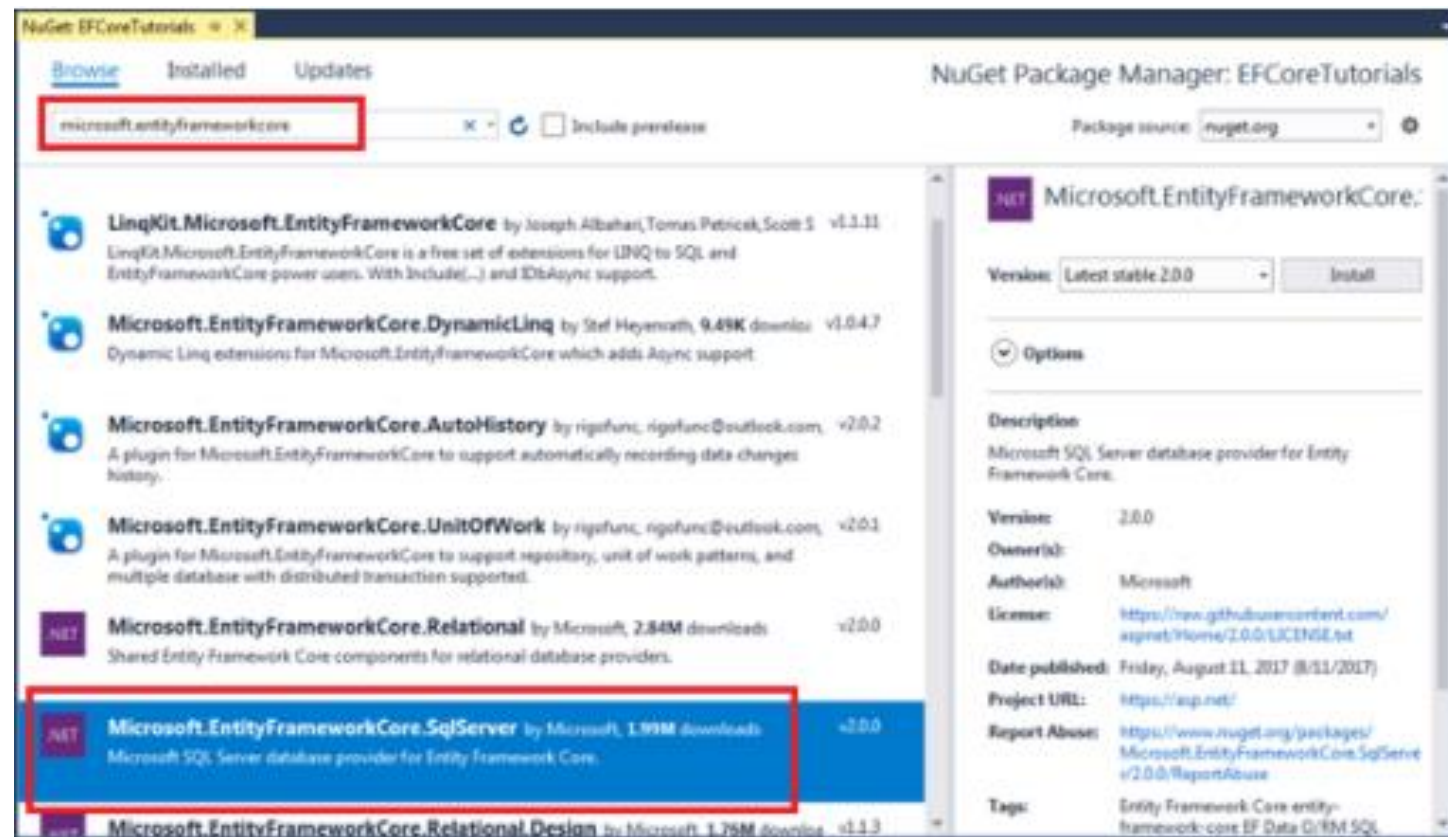
ột phải
à chọn
ols ->
es For

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

- Cài đặt trình cung cấp cơ sở dữ liệu của EF Core
 - Điều này sẽ mở giao diện trình quản lý gói **NuGet**. Nhấp vào tab **Browse** hoặc tab **Updates** và tìm kiếm **Microsoft.entityframeworkcore** trong hộp tìm kiếm ở góc trên cùng bên trái, như hình ảnh bên dưới.

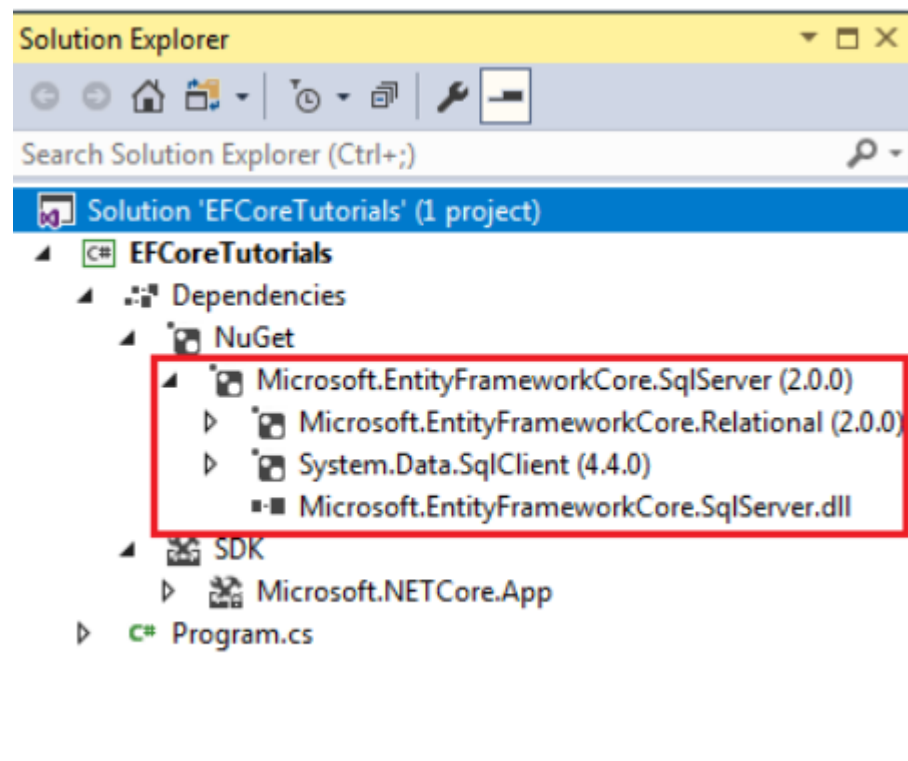


Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

- Cài đặt trình cung cấp cơ sở dữ liệu của EF Core
 - Sau khi cài đặt gói *Microsoft.EntityFrameworkCore.SqlServer*, kiểm tra nó trong **Dependencies** -> **NuGet**, như hình ảnh bên dưới.



1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

- **Cài đặt trình cung cấp cơ sở dữ liệu của EF Core**
 - Ngoài ra cũng có thể cài đặt gói **NuGet** của trình cung cấp bằng **Package Manager Console**. Chuyển đến **Tools -> NuGet Package Manager -> Package Manager Console** và thực hiện lệnh sau để cài đặt gói trình cung cấp **SQL Server**:

```
1  
2 PM> Install-Package Microsoft.EntityFrameworkCore.SqlServer
```

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

- **Cài đặt công cụ của EF Core**

- Cùng với gói trình cung cấp **DB**, cũng cần cài đặt các công cụ **EF** để thực thi các lệnh **EF Core**.
- Điều này giúp dễ dàng thực hiện một số nhiệm vụ liên quan đến **EF Core** trong dự án của bạn tại thời điểm thiết kế, chẳng hạn như chuyển đổi, scaffolding, v.v.
- Công cụ **EF** có sẵn dưới dạng các gói **NuGet**. Có thể cài đặt gói **NuGet** cho các công cụ **EF** tùy thuộc vào nơi bạn muốn thực thi các lệnh: sử dụng **Package Manager Console** (phiên bản PowerShell của các lệnh EF Core) hoặc sử dụng dotnet **CLI**.

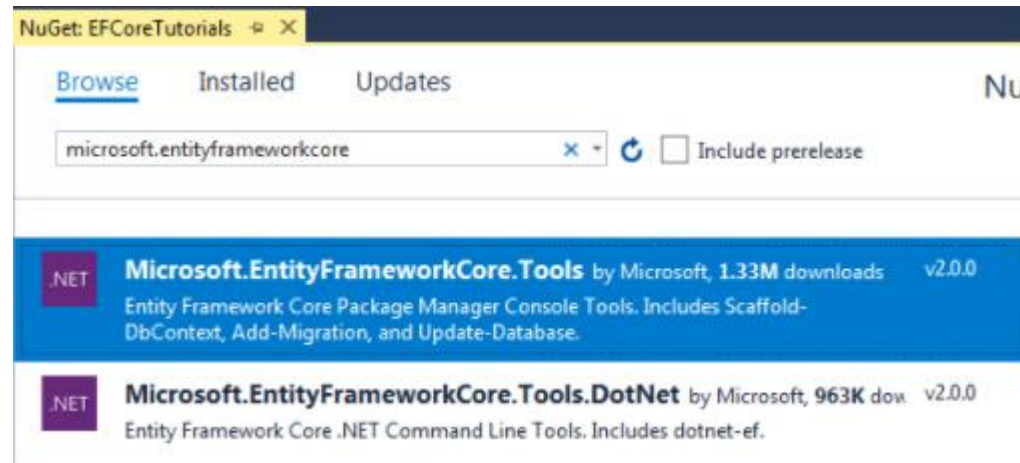
1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

- **Cài đặt công cụ của EF Core**

- **Cài đặt công cụ EF Core cho PMC**

- Để thực thi các lệnh **EF Core** từ **Package Manager Console**, hãy tìm kiếm gói **Microsoft.EntityFrameworkCore.Tools** từ **NuGet UI** và cài đặt nó như hình ảnh bên dưới.



- Điều này sẽ cho phép thực thi các lệnh **EF Core** cho scaffolding, chuyển đổi, vv trực tiếp từ **Package Manager Console (PMC)** trong Visual Studio.

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

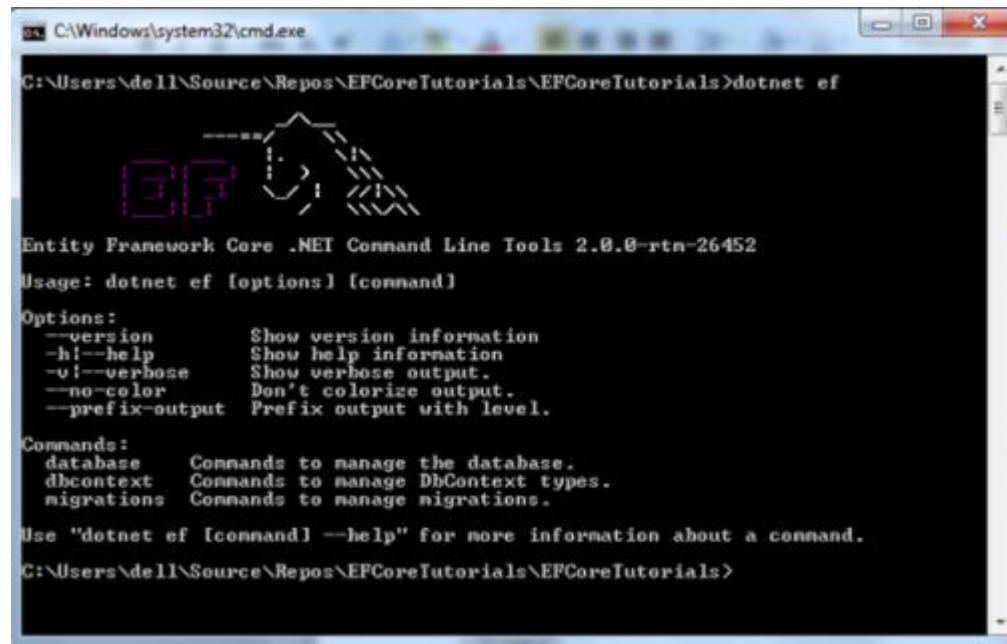
- **Cài đặt công cụ của EF Core**
 - **Cài đặt công cụ EF Core cho dotnet CLI**
 - Nếu muốn thực thi các lệnh **EF Core** từ **CLI** (Giao diện dòng lệnh) của **.NET Core**, trước tiên hãy cài đặt gói **NuGet *Microsoft.EntityFrameworkCore.Tools.DotNet*** bằng **NuGet UI**.
 - Sau khi cài đặt gói ***Microsoft.EntityFrameworkCore.Tools.DotNet***, chỉnh sửa file **.csproj** bằng cách nhấp chuột phải vào dự án trong **Solution Explorer** và chọn **Edit <projectname>.csproj**. Thêm node **<DotNetCliToolReference>** như ví dụ dưới đây. Đây là một bước bổ sung mà bạn cần thực hiện để thực thi các lệnh **EF Core 2.0** từ dotnet **CLI** trong **Visual Studio 2017**.

```
1  <Project Sdk="Microsoft.NET.Sdk">
2
3    <PropertyGroup>
4      <OutputType>Exe</OutputType>
5      <TargetFramework>netcoreapp2.0</TargetFramework>
6    </PropertyGroup>
7    <ItemGroup>
8      <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="2.0.0" />
9      <PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="2.0.0" />
10     <DotNetCliToolReference Include="Microsoft.EntityFrameworkCore.Tools.DotNet" Version="2.0.0" />
11   </ItemGroup>
12 </Project>
```

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

- Cài đặt công cụ của EF Core
 - Cài đặt công cụ EF Core cho dotnet CLI
 - Bây giờ, hãy mở dấu nhắc lệnh (hoặc thiết bị đầu cuối như PowerShell chẳng hạn) từ thư mục gốc của dự án của bạn và thực hiện các lệnh **EF Core** từ **CLI** bắt đầu bằng lệnh **dotnet ef**, như hình ảnh bên dưới.



```
C:\Windows\system32\cmd.exe
C:\Users\dell\Source\Repos\EFCoreTutorials\EFCoreTutorials>dotnet ef

Entity Framework Core .NET Command Line Tools 2.0.0-rtm-26452

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.
C:\Users\dell\Source\Repos\EFCoreTutorials\EFCoreTutorials>
```

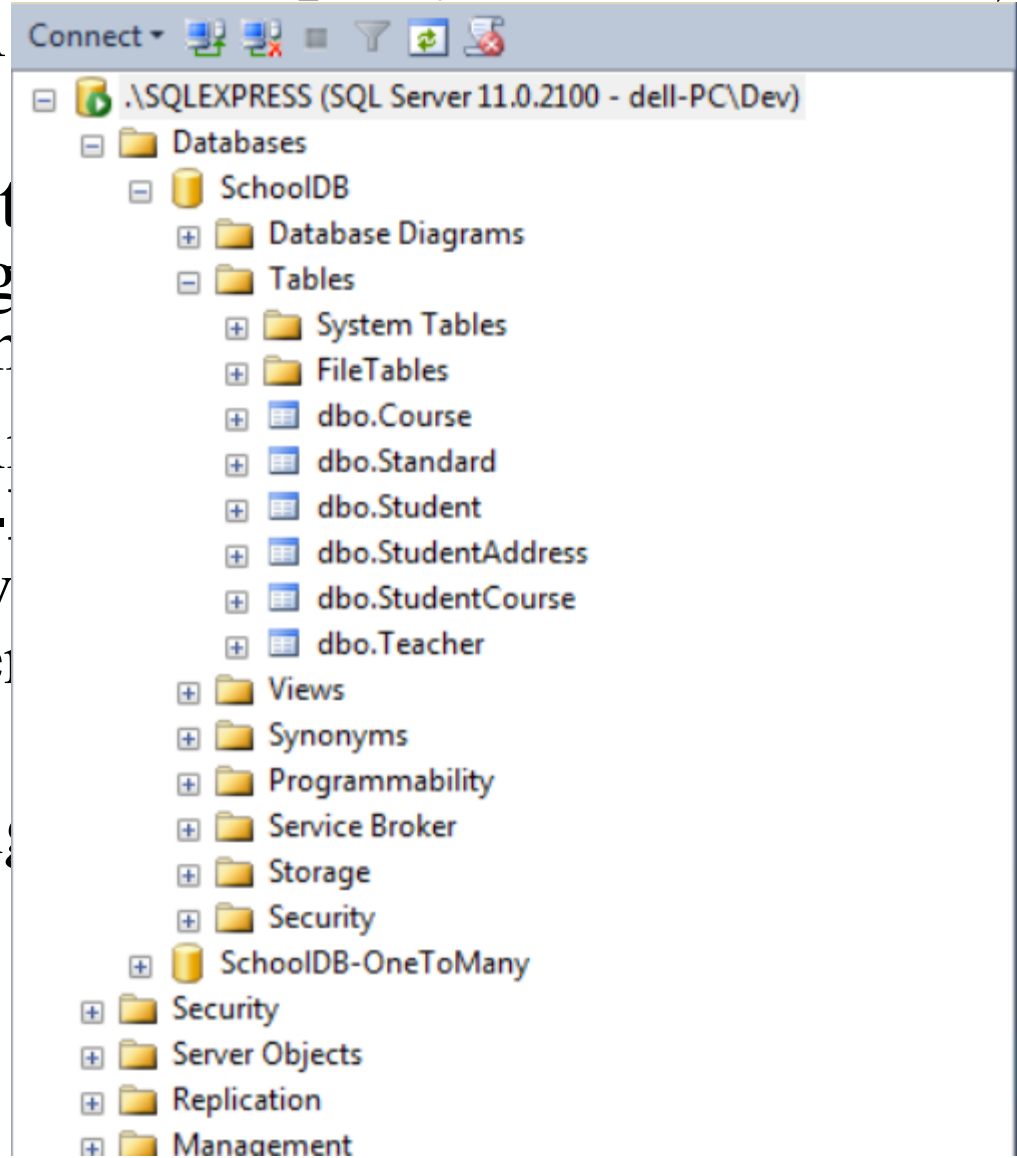

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

- Ở phần này, sẽ tìm hiểu cách tạo lớp **Context** và các lớp *thực thể* cho *cơ sở dữ liệu có sẵn* trong **Entity Framework Core**. Tạo các *lớp thực thể* và **Context** cho cơ sở dữ liệu có sẵn **Database First**.
- **EF Core** không hỗ trợ **DB** và trình hướng dẫn **Context** tương tự như **EF**.
- Vì vậy, cần thực hiện bằng lệnh **Scaffold-DbContext** để tạo ra các *lớp thực thể* và **Context** (gọi là **DbContext**) dựa trên database có sẵn.
- Tạo các *lớp thực thể* cho **SchoolDB** sau trong hình bên dưới.



1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

• Lệnh Scaffold-DbContext

- Sử dụng lệnh **Scaffold-DbContext** để tạo một mô hình dựa trên cơ sở dữ liệu hiện tại của bạn. Các tham số sau có thể được chỉ định cho **Scaffold-DbContext** trong **Package Manager Console**:

```
1 Scaffold-DbContext [-Connection] [-Provider] [-OutputDir] [-Context] [-Schemas>] [-Tables>]  
2 [-DataAnnotations] [-Force] [-Project] [-StartupProject] [<CommonParameters>]
```

- Trong **Visual Studio**, chọn menu Tools -> NuGet Package Manager -> Package Manager Console và chạy lệnh sau:

```
1 PM> Scaffold-DbContext "Server=.\SQLEXPRESS;Database=SchoolDB;Trusted_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
```

- Trong lệnh trên, tham số đầu tiên là một chuỗi kết nối bao gồm ba phần: *Máy chủ DB*, *tên cơ sở dữ liệu* và *thông tin bảo mật*.
- Ở đây, **Server=.\SQLEXPRESS**; đề cập đến máy chủ cơ sở dữ liệu **SQLEXPRESS** cục bộ. **Database=SchoolDB**; chỉ định tên cơ sở dữ liệu "SchoolDB" mà chúng ta sẽ tạo các lớp. **Trusted_Connection=True**; chỉ định xác thực **Windows**. Nó sẽ sử dụng thông tin đăng nhập **Windows** để kết nối với SQL Server.

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

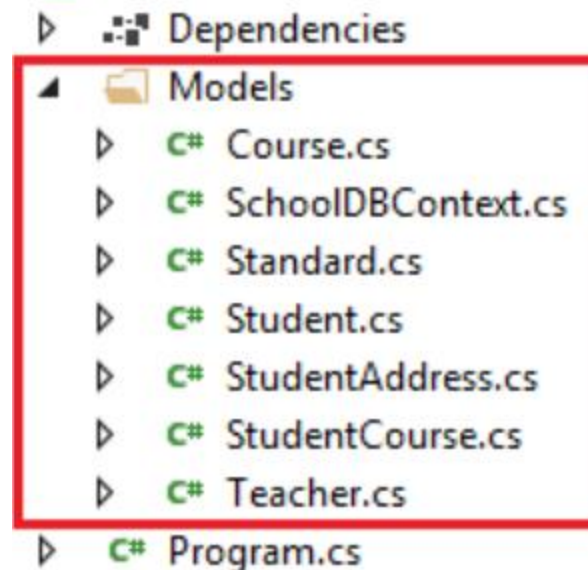
1.3. Tạo mô hình từ database có sẵn trong EF Core

• Lệnh Scaffold-DbContext

- Tham số thứ hai là tên nhà cung cấp. Sử dụng nhà cung cấp cho **SQL Server**, vì vậy nó là **Microsoft.EntityFrameworkCore.SqlServer**.
- Tham số **-OutputDir** mô tả thư mục nơi mà chúng tôi muốn tạo ra tất cả các lớp thực thể, đó là thư mục **Models**.
- Sử dụng lệnh sau để xem chi tiết về lệnh **Scaffold-DbContext**:

```
1 | PM> get-help scaffold-dbcontext -detailed
```

- Lệnh trên **Scaffold-DbContext** tạo các lớp thực thể cho mỗi bảng trong cơ sở dữ liệu **SchoolDB** và lớp **Context** với các cấu hình **Fluent API** cho 1  **EFCoreTutorials** ; thư mục **Models**.



1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

- **Lệnh Scaffold-DbContext**

- Sau đây là lớp thực thể **Student** được tạo cho bảng **Student**

```
1
2 using System;
3 using System.Collections.Generic;
4
5 namespace EFCoreTutorials.Models
6 {
7     public partial class Student
8     {
9         public Student()
10         {
11             StudentCourse = new HashSet<StudentCourse>();
12         }
13
14         public int StudentId { get; set; }
15         public string FirstName { get; set; }
16         public string LastName { get; set; }
17         public int? StandardId { get; set; }
18
19         public Standard Standard { get; set; }
20         public StudentAddress StudentAddress { get; set; }
21         public ICollection<StudentCourse> StudentCourse { get; set; }
22     }
23 }
```

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

- **Lệnh Scaffold-DbContext**

- Sau đây là lớp *SchoolDbContext* mà bạn có thể sử dụng để lưu hoặc truy xuất dữ liệu.

```
1 using System;
2 using Microsoft.EntityFrameworkCore;
3 using Microsoft.EntityFrameworkCore.Metadata;
4
5 namespace EFCoreTutorials.Models
6 {
7     public partial class SchoolDbContext : DbContext
8     {
9         public virtual DbSet<Course> Courses { get; set; }
10        public virtual DbSet<Standard> Standards { get; set; }
11        public virtual DbSet<Student> Students { get; set; }
12        public virtual DbSet<Teacher> Teachers { get; set; }
13        public virtual DbSet<Section> Sections { get; set; }
14        public virtual DbSet<Assignment> Assignments { get; set; }
15
16        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
17        {
18            if (!optionsBuilder.IsConfigured)
19            {
20                #warning To protect potentially sensitive information, this warning will appear if the connection
21                #warning string for the DbContext is not configured in your app's appsettings.json file.
22                #warning If you are not using multiple DbContexts within an In-Memory Database or on a test server,
23                #warning you should also call SetBaseDataContextAnnotation() to ensure consistent behavior.
24            }
25        }
26
27        protected override void OnModelCreating(ModelBuilder modelBuilder)
28        {
29            modelBuilder.Entity<Course>(entity =>
30            {
31                entity.Property(e => e.CourseName)
32                    .HasMaxLength(50)
33                    .IsUnicode(false);
34
35                entity.HasOne(d => d.Teacher)
36                    .WithMany(p => p.Course)
37                    .HasForeignKey(d => d.TeacherId)
38                    .OnDelete(DeleteBehavior.Cascade)
39                    .HasConstraintName("FK_Course_Teacher");
40            });
41
42            modelBuilder.Entity<Standard>(entity =>
43            {
44                entity.Property(e => e.Description)
45                    .HasMaxLength(50)
46                    .IsUnicode(false);
47
48                entity.Property(e => e.StandardName)
49                    .HasMaxLength(50)
50                    .IsUnicode(false);
51            });
52        }
53    }
54 }
```

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

- **DotNet CLI**

- Nếu bạn sử dụng giao diện dòng lệnh *dotnet* để thực thi các lệnh **EF Core** thì hãy mở dấu nhắc lệnh và điều hướng đến thư mục gốc của dự án và thực hiện lệnh *dotnet ef dbcontext scaffold* sau:

```
1 | > dotnet ef dbcontext scaffold "Server=.\SQLEXPRESS;Database=SchoolDB;Trusted_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer --o Models
```

- Như vậy là có thể tạo mô hình **EF Core** cho cơ sở dữ liệu có sẵn.

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

- Lớp **DbContext** là một phần không thể thiếu của **Entity Framework**. Một thể hiện của **DbContext** đại diện cho một phiên làm việc với cơ sở dữ liệu có thể được sử dụng để truy vấn và lưu các thể hiện của các thực thể của bạn vào cơ sở dữ liệu.
- **DbContext** là sự kết hợp của các mẫu **Unit Of Work** và **Repository**. **DbContext** trong **EF Core** cho phép chúng ta thực hiện các tác vụ sau:
 - Quản lý kết nối cơ sở dữ liệu.
 - Cấu hình mô hình và mối quan hệ.
 - Truy vấn cơ sở dữ liệu.
 - Lưu dữ liệu vào cơ sở dữ liệu.
 - Cấu hình theo dõi thay đổi.
 - Bộ nhớ đệm.
 - Quản lý giao dịch.

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

- **DbContext trong Entity Framework Core**

- Để sử dụng **DbContext** trong ứng dụng, chúng ta cần tạo lớp kế thừa từ lớp **DbContext**, còn được gọi là lớp **Context**.

```
public class SchoolContext : DbContext
{
    public SchoolContext()
    {
    }

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
    }

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
    }

    //entities
    public DbSet<Student> Students { get; set; }
    public DbSet<Course> Courses { get; set; }
}
```


1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

• DbContext trong Entity Framework Core

- Trong ví dụ trên, lớp *SchoolContext* có nguồn gốc từ lớp *DbContext* và chứa các thuộc tính *DbSet<Tentity>* của các lớp thực thể *Student* và *Course*. Nó cũng ghi đè hai phương thức *OnConfiguring* và *OnModelCreating*
- Chúng ta phải tạo một thể hiện của *SchoolContext* để kết nối với cơ sở dữ liệu và lưu hoặc truy xuất dữ liệu *Student* hoặc *Course*
- Phương thức *OnConfiguring()* cho phép chúng ta lựa chọn và cấu hình nguồn dữ liệu được sử dụng với một **Context** sử dụng *DbContextOptionsBuilder*.
- Phương thức *OnModelCreating()* cho phép chúng ta cấu hình mô hình sử dụng *ModelBuilder Fluent API*.

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

- **Các phương thức của DbContext**

Thuộc tính	Mô tả
ChangeTracker	Cung cấp truy cập vào thông tin và hoạt động cho các thể hiện của thực thể mà Context này đang theo dõi.
Database	Cung cấp truy cập vào thông tin và hoạt động liên quan đến cơ sở dữ liệu cho Context này.
Model	Trả về siêu dữ liệu của các thực thể, mối quan hệ giữa chúng và cách chúng ánh xạ tới cơ sở dữ liệu.

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

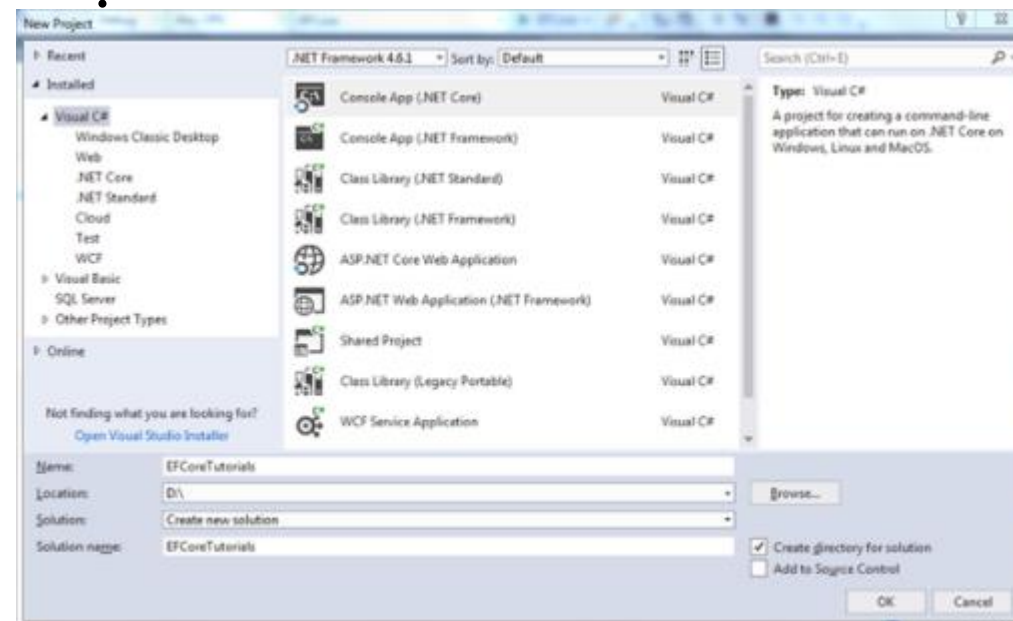
1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

- Ở hướng dẫn này, sẽ tìm hiểu từng bước cách sử dụng **Entity Framework Core** theo cách tiếp cận **Code First**. Để minh họa, sẽ tạo một ứng dụng **.NET Core Console** bằng **Visual Studio 2017** (hoặc mới hơn).
- Ứng dụng **.NET Core Console** có thể được tạo bằng **Visual Studio 2017** hoặc **Giao diện dòng lệnh (CLI)** cho **.NET Core**. Ở đây sẽ sử dụng **Visual Studio 2017**.
- Để tạo ứng dụng **.NET Core Console**, hãy mở **Visual Studio 2017** và chọn trên menu: **File -> New -> Project..** Hình ảnh bên dưới.



Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

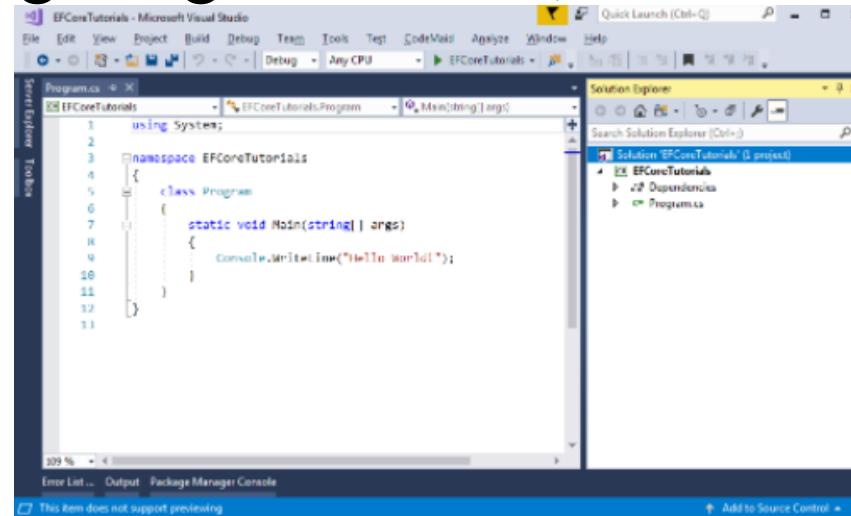
1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

- Trong cửa sổ New Project, mở rộng Installed -> Visual C# ở khung bên trái và chọn mẫu **Console App (.NET Core)** trong ngăn giữa.
- Nhập tên dự án, vị trí đặt dự án và nhấp vào nút **OK** để tạo ứng dụng bằng điều khiển, như hình ảnh bên dưới.



- Bây giờ, chúng ta cần cài đặt **EF Core** trong ứng dụng của mình bằng **Package Manager Console**. Chọn trên menu: Tools -> NuGet Package Manager -> Package Manager Console và thực hiện lệnh sau để cài đặt gói nhà cung cấp SQL Server:

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

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

• Tạo mô hình dữ liệu thực thể

- **Entity Framework** cần có một mô hình dữ liệu thực thể để giao tiếp với cơ sở dữ liệu. Nó xây dựng một mô hình dựa trên các lớp thực thể của bạn bằng cách sử dụng cấu hình chú thích dữ liệu và **Fluent API**.
- Mô hình **EF** bao gồm ba phần: mô hình khái niệm, mô hình lưu trữ và ánh xạ giữa mô hình khái niệm và mô hình lưu trữ.
- Theo cách tiếp cận **Code First**, **EF** xây dựng mô hình khái niệm dựa trên các lớp thực thể của bạn, lớp **Context** và các cấu hình.
- **EF Core** xây dựng mô hình lưu trữ và ánh xạ dựa trên trình cung cấp cơ sở dữ liệu bạn sử dụng. Ví dụ, mô hình lưu trữ của **SQL Server** sẽ khác với **DB2**.
- **EF** sử dụng mô hình này cho các hoạt động **CRUD** (**Create, Read, Update, Delete**) cho cơ sở dữ liệu.

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

• Tạo Migration trong EF Core

- **EF Core** có các lệnh chuyển đổi (**Migration**) khác nhau để tạo hoặc cập nhật cơ sở dữ liệu dựa trên mô hình.
- Tại thời điểm này chưa có cơ sở dữ liệu **SchoolDB**. Vì vậy, chúng ta cần tạo cơ sở dữ liệu từ mô hình (*thực thể* và *Context*) bằng cách thêm chuyển đổi.
- Chúng ta có thể thực thi lệnh chuyển đổi bằng cách sử dụng **Package Manger Console** hoặc **dotnet CLI** (giao diện dòng lệnh).
- Trong **Visual Studio**, mở **NuGet Package Manager Console** từ Tools -> NuGet Package Manager -> Package Manager Console và nhập lệnh sau:

add-migration CreateSchoolDB

- Nếu bạn sử dụng **dotnet CLI**, hãy nhập lệnh sau.

> dotnet ef migrations add CreateSchoolDB

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

- Các quy ước là các quy tắc mặc định bằng cách sử dụng **Entity Framework** để xây dựng một mô hình dựa trên các lớp thực thể của bạn.
- **EF Core API** tạo ra một lược đồ cơ sở dữ liệu dựa trên các *lớp thực thể* và lớp **Context**, không có bất kỳ cấu hình bổ sung nào vì các lớp thực thể đã tuân theo các quy ước.
- Hãy xem các *lớp thực thể* và lớp **Context** sau để hiểu các quy ước mặc định.

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

```
public class Student
{
    public int StudentId { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime DateOfBirth { get; set; }
    public byte[] Photo { get; set; }
    public decimal Height { get; set; }
    public float Weight { get; set; }

    public int GradeId { get; set; }
    public Grade Grade { get; set; }
}

public class Grade
{
    public int Id { get; set; }
    public string GradeName { get; set; }
    public string Section { get; set; }

    public IList<Student> Students { get; set; }
}

public class SchoolContext : DbContext
{
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlServer(@"Server=.\SQLEXPRESS;Database=SchoolDB;Trusted_Connection=True;");
    }

    public DbSet<Student> Students { get; set; }
}
```

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

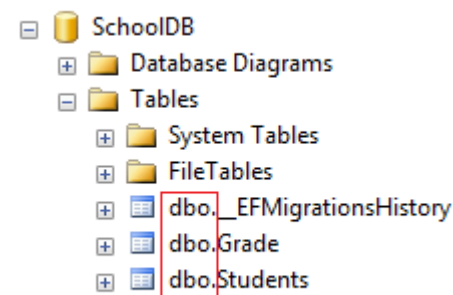
1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

• Lược đồ

- Theo mặc định, **EF Core** sẽ tạo tất cả các đối tượng cơ sở dữ liệu trong lược đồ **dbo**.



• Bảng

- EF Core** sẽ tạo các bảng cơ sở dữ liệu cho tất cả các thuộc tính *DbSet<TEntity>* trong lớp *Context* có cùng tên với thuộc tính.
- Nó cũng sẽ tạo các bảng cho các thực thể không phải là thuộc tính *DbSet* nhưng có thể truy cập thông qua các thuộc tính tham chiếu trong các thực thể *DbSet* khác.
- Đối với ví dụ trên, EF Core sẽ tạo bảng **Students** cho thuộc tính *DbSet<Student>* trong lớp *SchoolContext* và bảng **Grade** cho thuộc tính **Grade** trong lớp thực thể **Student**, mặc dù lớp *SchoolContext* không có thuộc tính *DbSet<Grade>*

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

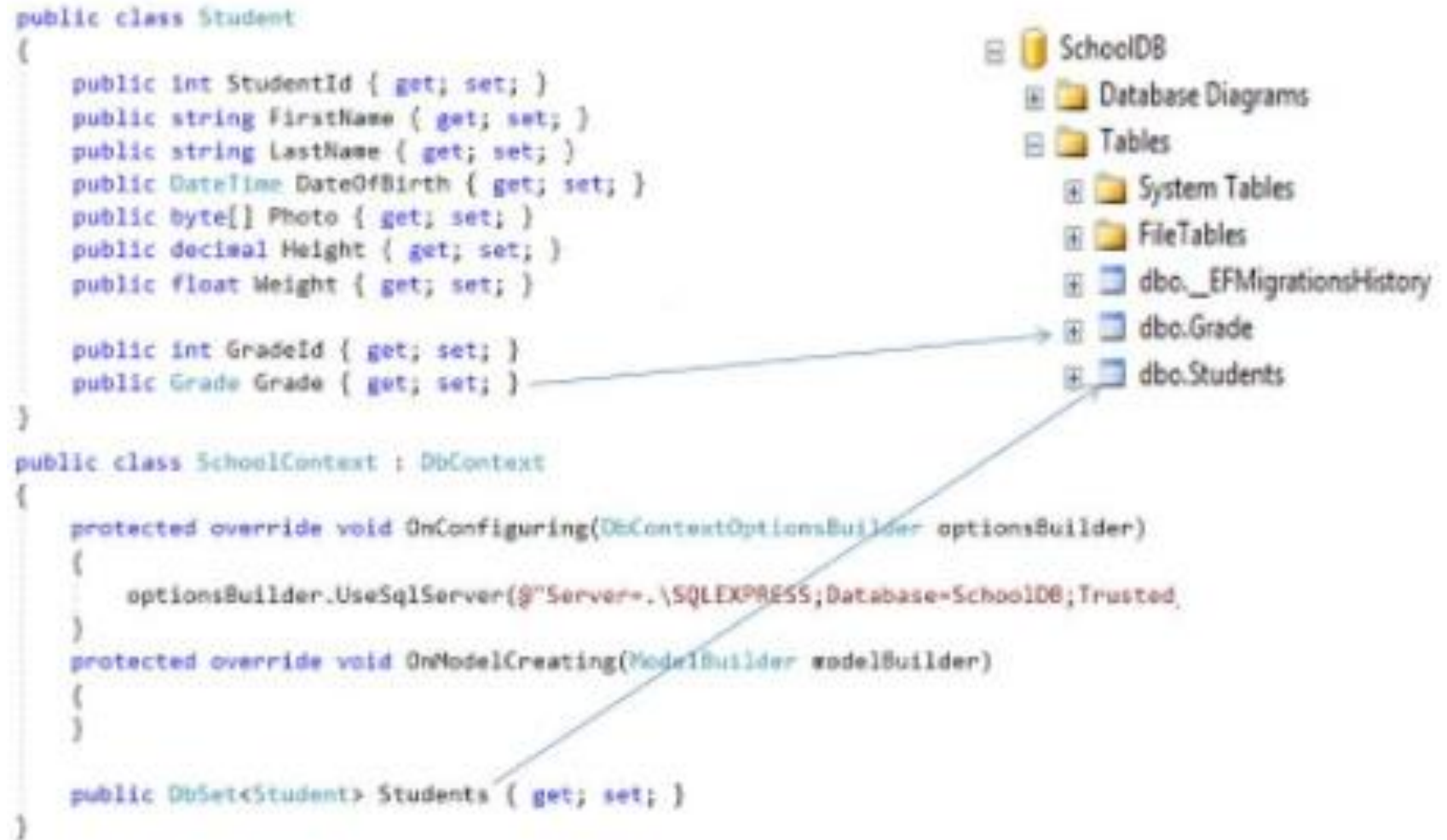
1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

• Bảng



Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

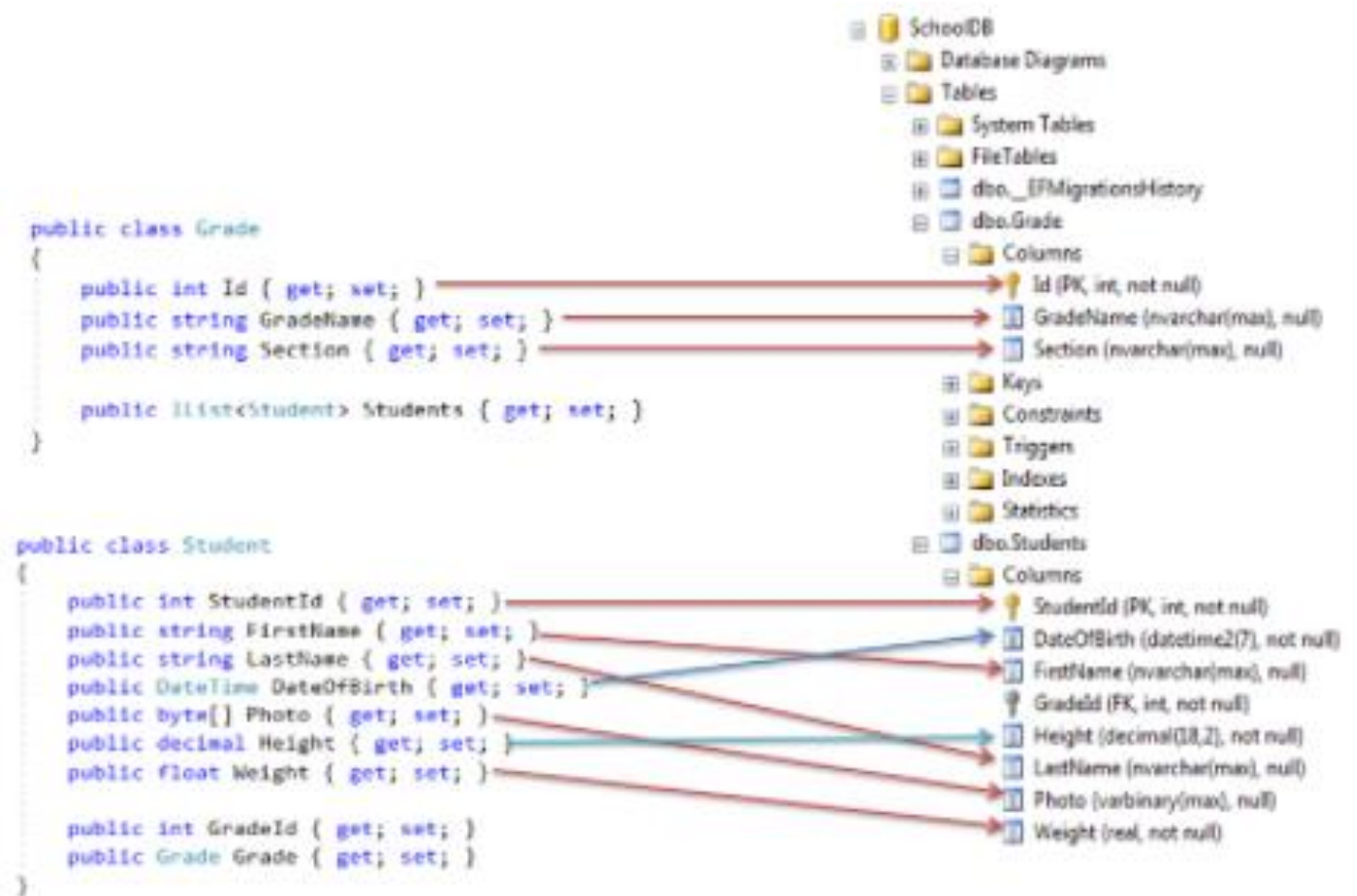
1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

• Cột

- Theo mặc định, **EF Core** sẽ tạo các cột cho tất cả các thuộc tính của một lớp thực thể có cùng tên với thuộc tính.
- Nó sử dụng các thuộc tính tham chiếu và thuộc tính tập hợp trong việc xây dựng mối quan hệ giữa các bảng tương ứng trong cơ sở dữ liệu.



1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

• Kiểu dữ liệu cột

- Kiểu dữ liệu cho các cột trong bảng cơ sở dữ liệu tùy thuộc vào cách nhà cung cấp cơ sở dữ liệu ánh xạ kiểu dữ liệu C# với kiểu dữ liệu của cơ sở dữ liệu đã chọn.

Kiểu dữ liệu C#	Ánh xạ tới kiểu dữ liệu SQL Server		
int	int	double	float
string	nvarchar(Max)	char	Không có ánh xạ
decimal	decimal(18,2)	sbyte	Không có ánh xạ (ném ra exception)
float	real	object	Không có ánh xạ
byte[]	varbinary(Max)		
datetime	datetime		
bool	bit		
byte	tinyint		
short	smallint		
long	bigint		

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

- **Cột Null**

- EF Core tạo các cột **null** cho tất cả các kiểu dữ liệu tham chiếu và các thuộc tính kiểu nguyên thủy **nullable**, ví dụ: *string*, *Nullable<int>*, *decimal?*.

- **Cột NotNull**

- EF Core tạo các cột **NotNull** trong cơ sở dữ liệu cho tất cả các thuộc tính khóa chính và các thuộc tính kiểu nguyên thủy, ví dụ: *int*, *float*, *decimal*, *DateTime*, v.v.

- **Khóa chính**

- EF Core sẽ tạo cột khóa chính cho thuộc tính có tên **Id** hoặc *<Entity Class Name>Id* (không phân biệt chữ hoa chữ thường).
- Ví dụ, EF Core sẽ tạo một cột khóa chính trong bảng *Students* nếu lớp *Student* có một thuộc tính có tên *id*, *ID*, *iD*, *Id*, *studentid*, *StudentId*, *STUDENTID* hoặc *sTUDENTID*.

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

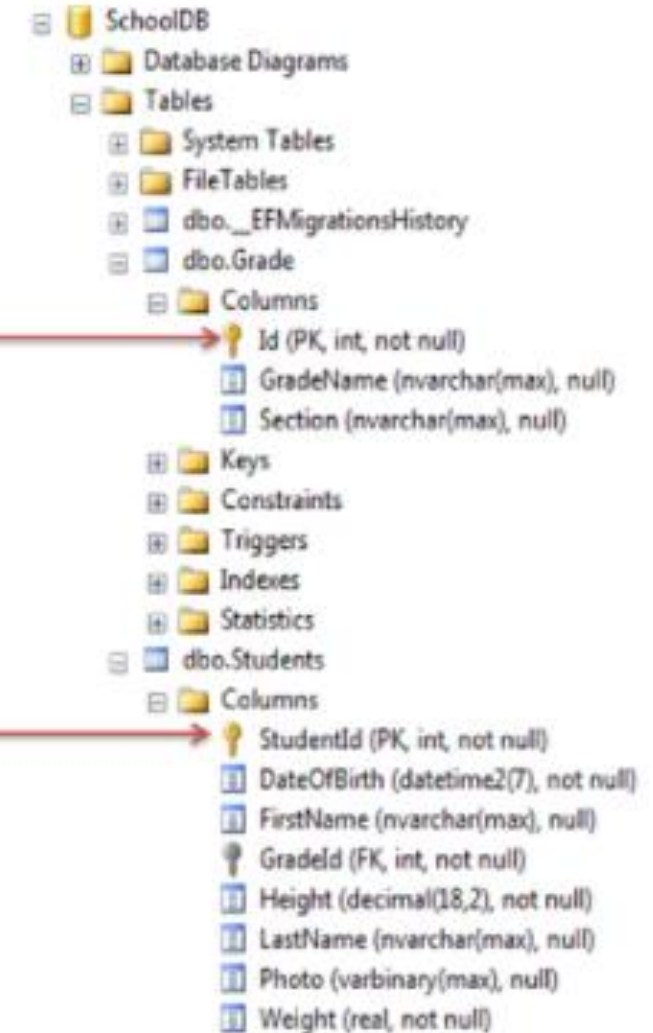
1.6. Các quy ước trong Entity Framework Core

```
public class Grade
{
    public int Id { get; set; }
    public string GradeName { get; set; }
    public string Section { get; set; }

    public IList<Student> Students { get; set; }
}
```

```
public class Student
{
    public int StudentId { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime DateOfBirth { get; set; }
    public byte[] Photo { get; set; }
    public decimal Height { get; set; }
    public float Weight { get; set; }

    public int GradeId { get; set; }
    public Grade Grade { get; set; }
}
```



Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

• Khóa ngoại

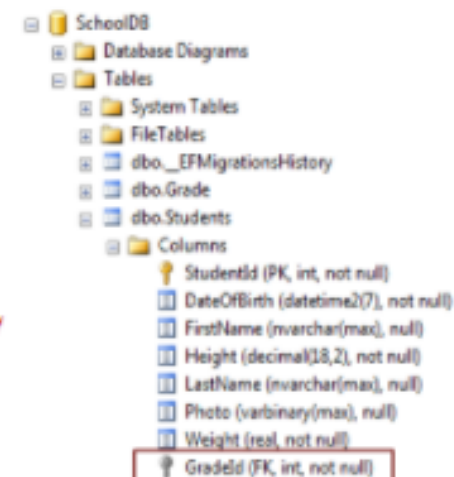
- Theo quy ước khóa ngoại, **EF Core API** sẽ tạo một cột khóa ngoại cho mỗi thuộc tính điều hướng tham chiếu trong một thực thể với một trong các mẫu đặt tên sau.
 - <Reference Navigation Property Name>Id
 - <Reference Navigation Property Name><Principal Primary Key Property Name>
- Trong ví dụ (các thực thể *Student* và *Grade*), **EF Core** sẽ tạo một cột khóa ngoại *GradeId* trong bảng *Students*, như được mô tả trong hình dưới đây.

```
public class Student Dependent Entity
{
    public int StudentId { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime DateOfBirth { get; set; }
    public byte[] Photo { get; set; }
    public decimal Height { get; set; }
    public float Weight { get; set; }

    public int GradeId { get; set; } Foreign Key Property
    public Grade Grade { get; set; } Reference Property
}

public class Grade Principal Entity
{
    public int Id { get; set; } Primary Key Property
    public string GradeName { get; set; }
    public string Section { get; set; }

    public IList<Student> Students { get; set; }
}
```



1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

- **Khóa ngoại**

- Bảng sau liệt kê các tên cột khóa ngoại cho các tên thuộc tính tham chiếu khác và các tên thuộc tính khóa chính.

Tên thuộc tính tham chiếu trong thực thể phụ thuộc	Tên thuộc tính khóa ngoại trong thực thể phụ thuộc	Tên thuộc tính khóa chính	Tên cột khóa ngoại trong DB
Grade	GradeId	GradeId	GradeId
Grade	-	GradeId	GradeId
Grade	-	Id	GradeId
CurrentGrade	CurrentGradeId	GradeId	CurrentGradeId
CurrentGrade	-	GradeId	CurrentGradeGradeId
CurrentGrade	-	Id	CurrentGradeId
CurrentGrade	GradeId	Id	GradeId

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

- **Index**

- Theo mặc định, **EF Core** tạo *clustered index* trên các cột khóa chính và *non-clustered index* trên các cột khóa ngoại.

- **Quy ước về mối quan hệ một-nhiều trong EF Core**

- Ở phần này, sẽ tìm hiểu về các quy ước mối quan hệ giữa hai lớp thực thể dẫn đến mối quan hệ một-nhiều giữa các bảng tương ứng trong cơ sở dữ liệu.
- **Entity Framework Core** tuân theo quy ước tương tự như các quy ước mối quan hệ một-nhiều trong **Entity Framework 6.x**. Sự khác biệt duy nhất là **EF Core** tạo cột khóa ngoại có cùng tên với tên thuộc tính điều hướng chứ không phải là *<NavigationPropertyName>_<PrimaryKeyPropertyName>* như trong **Entity Framework**.

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

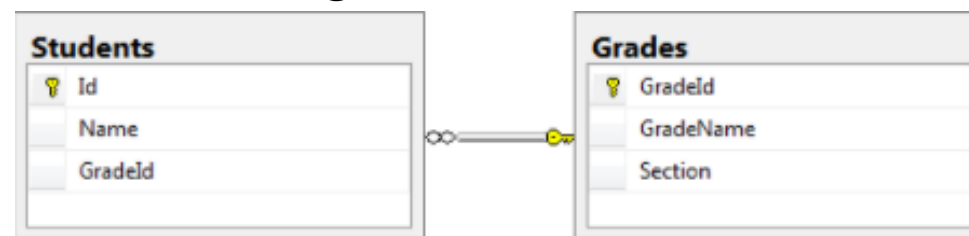
1.6. Các quy ước trong Entity Framework Core

• Quy ước về mối quan hệ một-nhiều trong EF Core

- Chúng ta sẽ xem các quy ước khác nhau trong **EF Core** tự động cấu hình mối quan hệ một-nhiều giữa các thực thể *Student* và *Grade*

```
1
2 public class Student
3 {
4     public int StudentId { get; set; }
5     public string StudentName { get; set; }
6 }
7
8 public class Grade
9 {
10    public int GradeId { get; set; }
11    public string GradeName { get; set; }
12    public string Section { get; set; }
13 }
```

- Sau khi áp dụng các quy ước cho *mỗi quan hệ một-nhiều* trong các thực thể ở trên, các bảng cơ sở dữ liệu cho các thực thể *Student* và *Grade* sẽ trông giống như hình bên dưới, trong đó bảng *Students* có khóa ngoại *GradeId*



Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

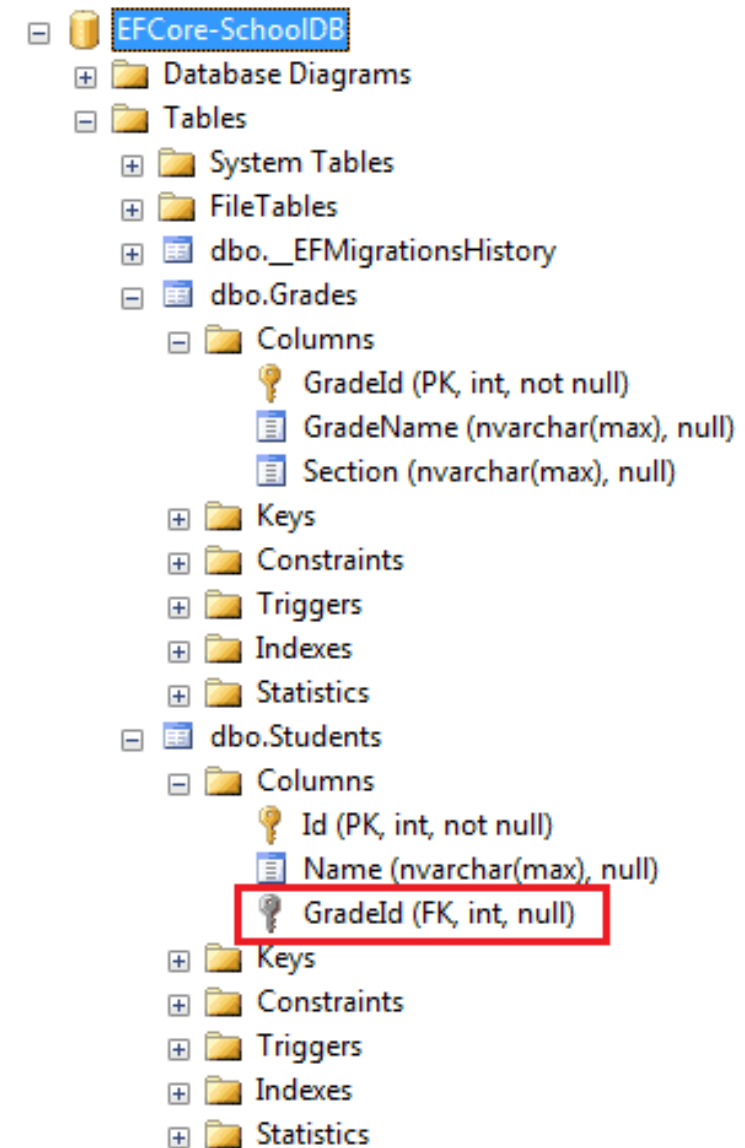
1.6. Các quy ước trong Entity Framework Core

- Quy ước về mối quan hệ một-nhiều trong EF Core
 - Cách 1:

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }

    public Grade Grade { get; set; }
}

public class Grade
{
    public int GradeId { get; set; }
    public string GradeName { get; set; }
    public string Section { get; set; }
}
```



1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

- Quy ước về mối quan hệ một-nhiều trong EF Core
 - Cách 2:

```
public class Student
{
    public int StudentId { get; set; }
    public string StudentName { get; set; }
}

public class Grade
{
    public int GradeId { get; set; }
    public string GradeName { get; set; }
    public string Section { get; set; }

    public ICollection<Student> Students { get; set; }
}
```

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

- Quy ước về mối quan hệ một-nhiều trong EF Core
 - Cách 3:

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }

    public Grade Grade { get; set; }
}

public class Grade
{
    public int GradeID { get; set; }
    public string GradeName { get; set; }

    public ICollection<Student> Students { get; set; }
}
```

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

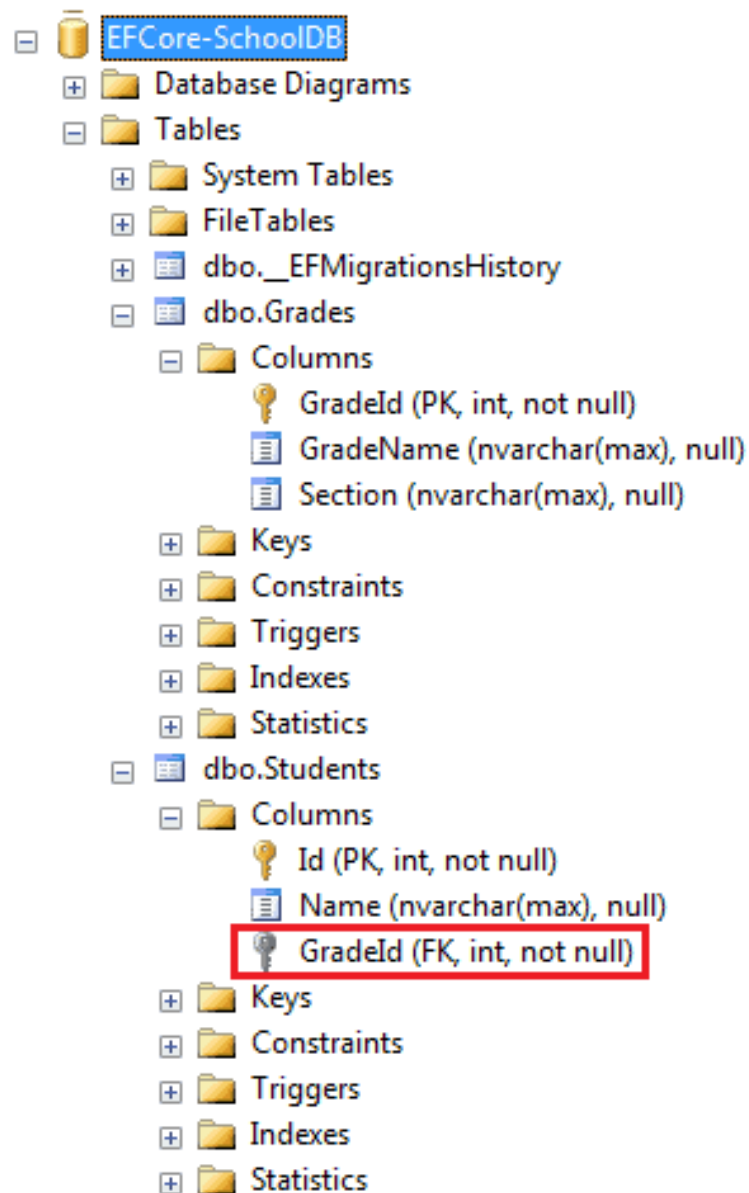
- Quy ước về mối quan hệ một-nhiều trong EF Core
 - Cách 4:

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }

    public int GradeId { get; set; }
    public Grade Grade { get; set; }
}

public class Grade
{
    public int GradeId { get; set; }
    public string GradeName { get; set; }

    public ICollection<Student> Students { get; set; }
}
```



Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

• Quy ước về mối quan hệ một-một trong EF Core

- Entity Framework Core đã giới thiệu các quy ước mặc định tự động định cấu hình *mối quan hệ một-một* giữa hai thực thể (**EF 6.x** hoặc trước đó không hỗ trợ các quy ước cho mối quan hệ một-một).
- Trong **EF Core**, *mối quan hệ một-một* đòi hỏi một thuộc tính điều hướng tham chiếu ở cả hai bên. Sau đây các thực thể *Student* và *StudentAddress* tuân theo quy ước cho *mối quan hệ một-một*.

```
1
2 public class Student
3 {
4     public int Id { get; set; }
5     public string Name { get; set; }
6
7     public StudentAddress Address { get; set; }
8 }
9
10 public class StudentAddress
11 {
12     public int StudentAddressId { get; set; }
13     public string Address { get; set; }
14     public string City { get; set; }
15     public string State { get; set; }
16     public string Country { get; set; }
17
18     public int StudentId { get; set; }
19     public Student Student { get; set; }
20 }
```

Chương 1. Tổng quan về Entity Framework Core

1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

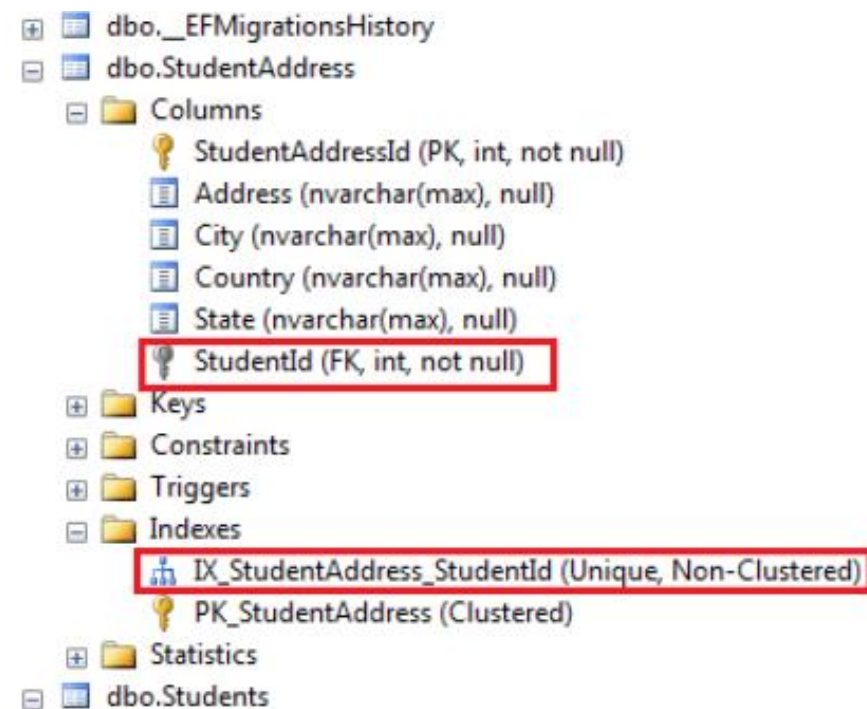
1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

• Quy ước về mối quan hệ một-một trong EF Core

- Trong ví dụ trên, thực thể *Student* có một thuộc tính điều hướng tham chiếu thuộc kiểu *StudentAddress* và thực thể *StudentAddress* có một thuộc tính khóa ngoại *StudentId* và thuộc tính tham chiếu tương ứng của nó là *Student*
- Điều này sẽ dẫn đến *mối quan hệ một-một* trong các bảng tương ứng *Students* và *StudentAddresses* trong cơ sở dữ liệu, như được hiển thị trong hình bên dưới.



1.1. Giới thiệu Entity Framework Core

1.2. Cài đặt Entity Framework Core

1.3. Tạo mô hình từ database có sẵn trong EF Core

1.4. DbContext trong EF Core

1.5. Tạo ứng dụng EF Core đầu tiên

1.6. Các quy ước trong Entity Framework Core

• Quy ước về mối quan hệ một-một trong EF Core

- EF Core tạo một *unique index* trên cột khóa ngoại *StudentId* *NotNull* trong bảng *StudentAddresses*, như được hiển thị ở trên.
- Điều này đảm bảo rằng giá trị của cột khóa ngoại *StudentId* phải là duy nhất trong bảng *StudentAddress*, điều cần thiết cho *mối quan hệ một-một*.

Thank you!