



ASP .NET Core 3.0 {پیشرفته}

 Visual Studio 2019

مدرس: علیرضا ارومند
nikamooz;
آموزش برنامه نویسی و اجرای پروژه

www.NikAmooz.com

جلسه نهم

معرفی علیرضا ارومند

۱. مدرس و مشاور ASP.NET Core و معماری‌های نرم‌افزاری (نیک آموز)
۲. مدیر فنی خبرگزاری نسیم
۳. کارشناس ارشد توسعه نرم افزار داتین (فناپ)
۴. کارشناس ارشد توسعه نرم افزار ارتباط فردا (بانک آینده)
۵. متخصص انجام پروژه‌های وب و .NET
۶. و ...



Configuring non-relational properties

مدرس: علیرضا ارومند

چه خواهیم آموخت

۱. انواع روش‌های Configuration
۲. آشنایی با تنظیمات non-relational
۳. Shadow Property چیست؟
۴. Backing-field چیست؟



سه روش انجام تنظیمات



۱. استفاده از قراردادها

۲. استفاده از Data Annotation

۳. تنظیمات با Fluent API

استفاده از قراردادها



۱. اعمال به صورت پیش فرض
۲. باز نویسی با دو روش دیگر

قرار دادهای Entity Class

۱. کلاس باید Public باشد

۲. نمی‌تواند Static باشد

۳. باید سازنده پیش فرض داشته باشد



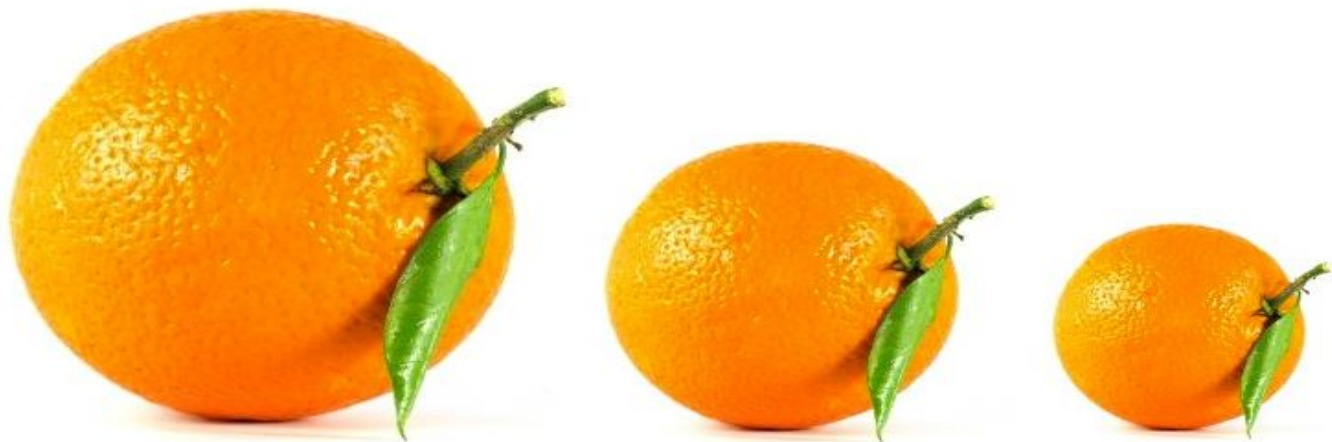
قرار دادهای خواص Entity

۱. خواص باید Public باشد
۲. Getter باید Public باشد
۳. Setter می‌تواند Public نباشد



قرار دادهای نام، نوع و سایز

۱. نام خاصیت و ستون جدول یکسان
۲. تبدیل انواع Net. توسط Provider
۳. DateTime به DateTime2



Allow Null



۱. رشته اجازه خالی بودن دارد

۲. در Primitive Type ها امکان ندارد

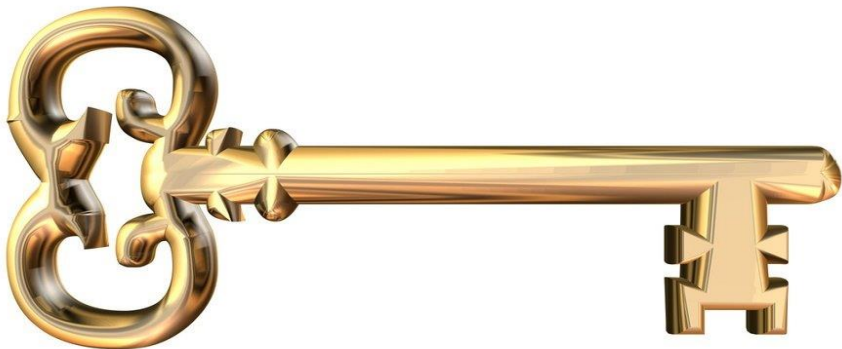
۳. Nullable type های اجازه null دارند

کلید اصلی

۱. پیش فرض Composite Key نیست

۲. نام Id یا Class Name+Id

۳. نوع کلید تولید کننده اصلی را تعیین می کند

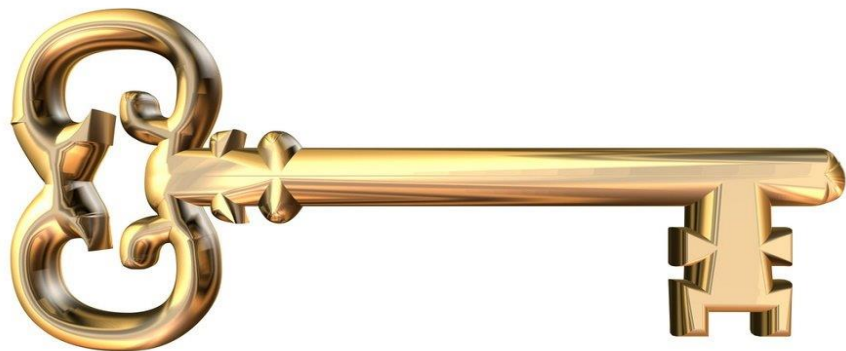


تولید کننده کلید

۱. مقادیر عددی پایگاه داده

۲. GUID توسط EF

۳. مقادیر رشته‌ای توسط برنامه نویس



پیشنه‌ای

با اینکه Id مختصرتر و نوشتن کد ساده‌تر است، جهت خوانایی بیشتر کد نویسی از **Class Name + Id** استفاده کنید.

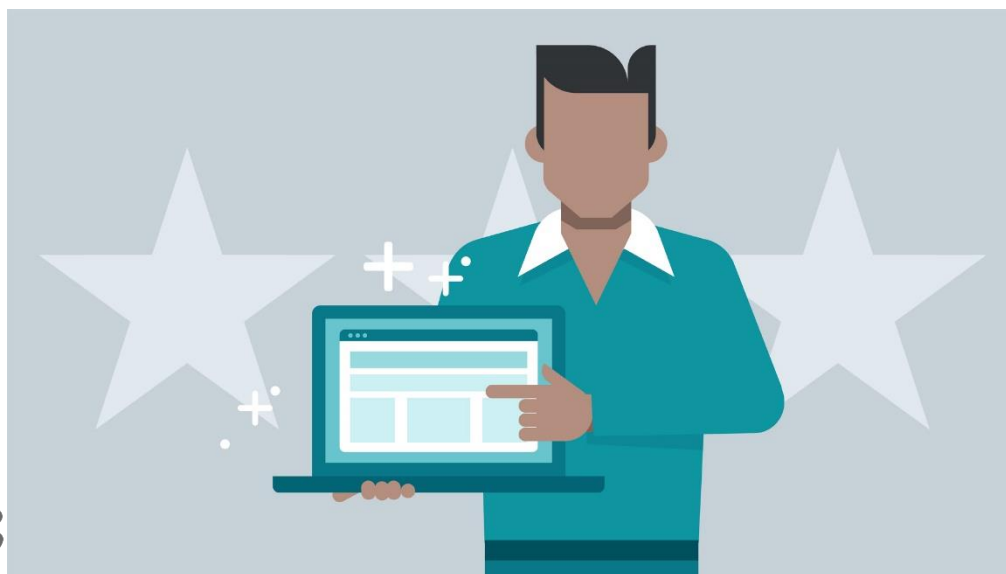


تنظیمات به کمک Annotation

۱. پیاده سازی به کمک Attribute ها

۲. قابل استفاده در سطح کلاس و خواص

۳. اصطلاحاً Data Annotation می نامیم



System.ComponentModel.DataAnnotations

۱. فضای نام اصلی برای Annotation

۲. در Front-End هم کاربرد دارد

۳. برخی از Attribute ها بی تاثیر در DB



فضای نام Schema



۱. تخصصی برای دیتابیس

۲. نام جدول، ستون و ...

استفاده از Fluent Validation



۱. می‌توان در `OnModelCreating` نوشت
۲. جدا سازی با `IEntityTypeConfiguration`
۳. جایگزین `EntityTypeConfiguration`

چه زمانی اجرا می‌شوند

۱. در زمان ایجاد اولین نمونه

۲. هنگام اجرای تابع OnModelCreating

۳. اجرا به ترتیب توضیح



ممانعت از ایجاد جدول و ستون

۱. به کمک خاصیت `NotMapped`

۲. به کمک `Ignore` در روش `Fluent`



آشنایی با برخی Attribute ها

عملکرد	نام Attribute
تعیین اجباری بودن مقدار	[Required]
تعیین حداکثر طول برای رشته	[MaxLength(123)]
تعیین جنس و اندازه دیتابیس	[Column(DataType = "---")]

معادل IsRequired، HasMaxLength و HasColumnType در روش Fluent

تنظیمات Primary Key



۱. عدم رعایت Convention

۲. نیاز به شناسه ترکیبی

۳. استفاده از Key در Data Annotation

تنظیمات Primary Key



۴. تعیین ترتیب به کمک Column لازم

۵. استفاده از HasKey در Fluent

ایندکس گذاری با EF

۱. فقط با Fluent API امکان دارد

۲. تعریف با `HasIndex`

۳. نام ایندکس با `HasName`

۴. یکتا بودن با `IsUnique`



تعیین نام برای عناصر دیتابیس



۱. تفاوت قواعد نام گذاری DB با App

۲. نام جدول با Table در Attribute

۳. استفاده از ToTable در Fluent

تعیین نام برای عناصر دیتابیس



۴. نام ستون با Column در Fluent

۵. استفاده از HasColumnName در Fluent

Schema



۱. تنظیم توسط دیتابیس انجام می شود
۲. عدم پشتیبانی در برخی Provider ها
۳. HasDefaultSchema تعیین پیش فرض
۴. تعیین همراه با تعیین نام جدول

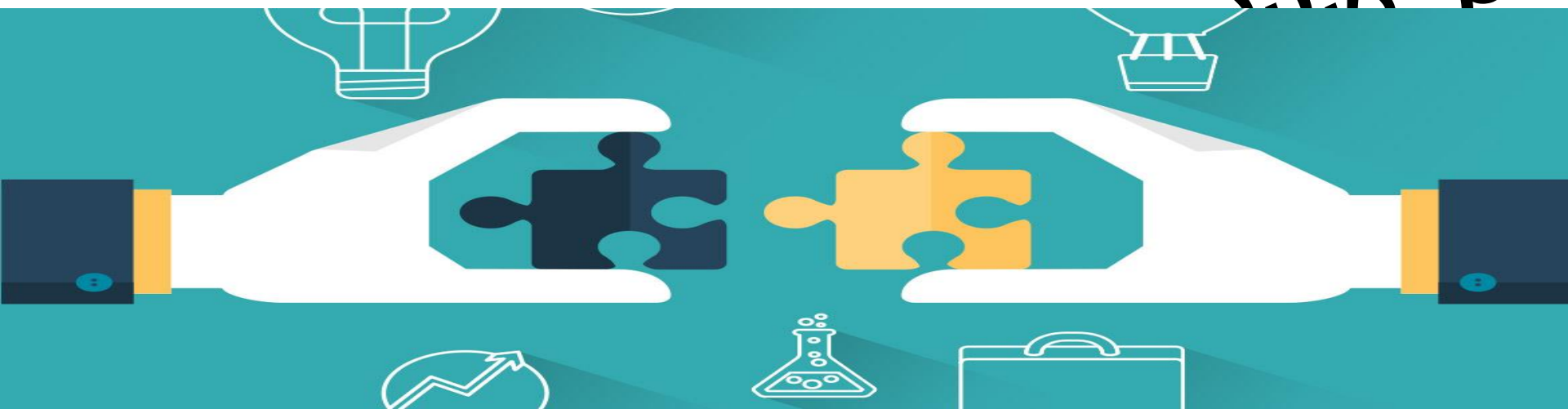
تنظیمات مخصوص Provider ها

۱. همه Provider ها یکسان نیستند

۲. Provider ها متد الحاقی شناسایی دارند

۳. IsSqlServer و IsSqlite نمونه ای از این
متدها

۴. کلاس



کدام روش را استفاده کنیم؟

۱. تا حد امکان استفاده از Convention
۲. در صورت نیاز سایر قسمت‌ها Attribute
۳. Fluent API کاملترین و منعطفترین



Shadow Properties

۱. از امکانات جدید در Ef Core

۲. ایجاد ستون بدون معادل در برنامه



Shadow Properties

۳. نیاز به معرفی نوع و نام ستون داریم

۴. انجام به کمک Property در Fluent



Shadow Properties

۱. نباید خاصیت هم‌نام وجود داشته باشد
۲. امکان تفاوت نام خاصیت و ستون



Shadow Properties

۳. `Entity(myEntity).Property
("MyPropertyName").CurrentValue`
۴. فقط با `Tracked Data` کار می‌کند



Backing Field



۱. معرفی در EF Core

۲. امکان استفاده از Private Field

۳. روشی برای کنترل دسترسی به خواص

Value Conversion



۱. امکان ایجاد تغییر پیش از ذخیره
۲. امکان جدید در EF Core 2.1

کاربردهای متداول

۱. تبدیل نوع Enum ها
۲. تبدیل کلاس به `string` و بالعکس
۳. کدگذاری اتوماتیک



Configuring relationships

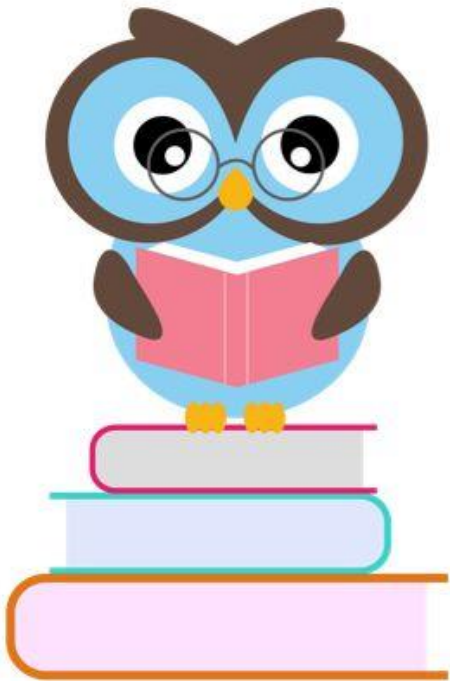
مدرس: علیرضا ارومند

چه خواهیم آموخت

۱. تنظیمات پیش فرض روابط

۲. تنظیم روابط با `note annotation`

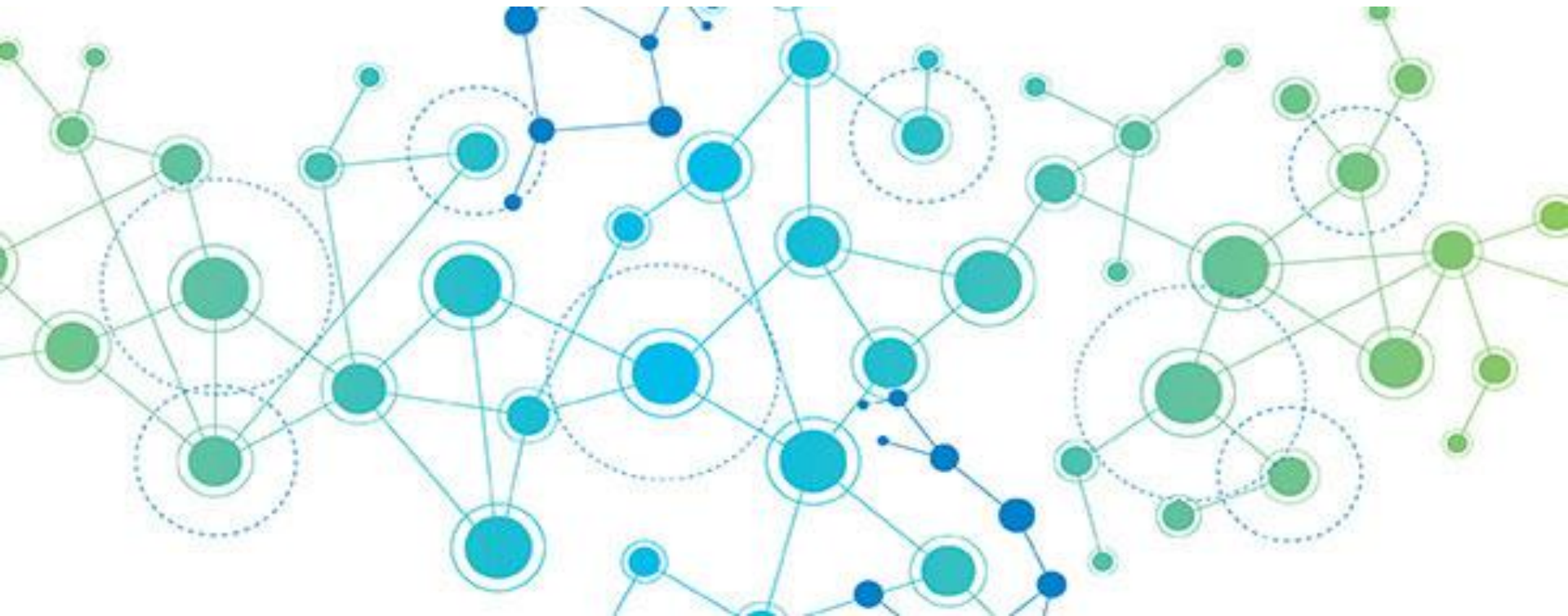
۳. ...



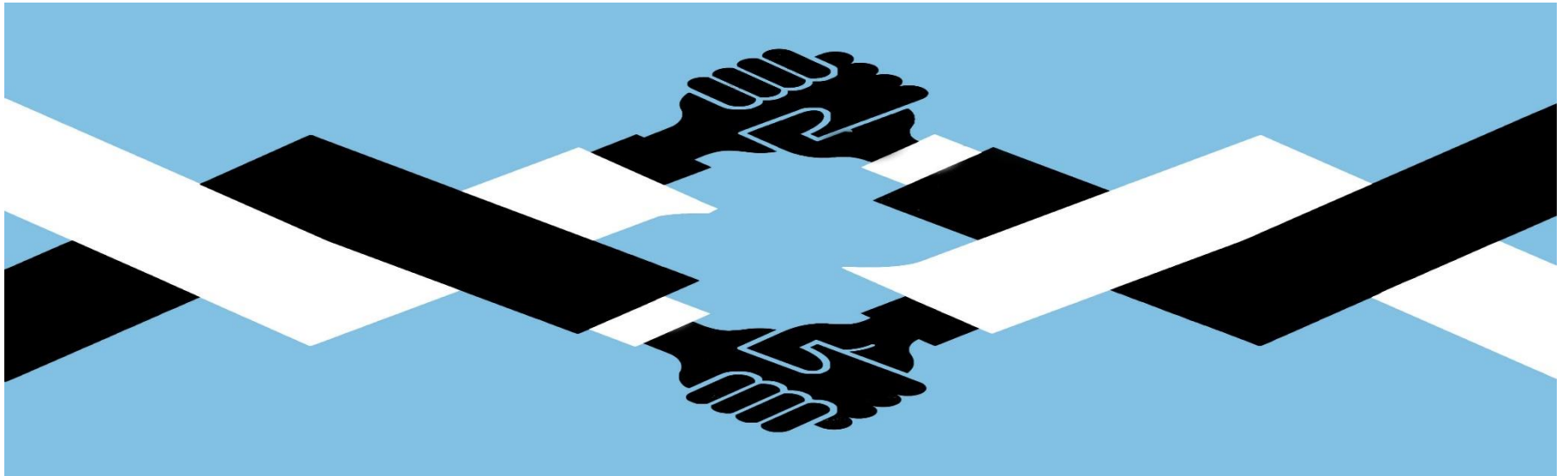
چیست؟ Navigation Property

۱. خاصیتی از جنس Entity دیگر

۲. میتواند لیست یا تکی باشد



چه زمانی نیاز داریم؟



۱. fully defined relationship رابطه دو طرفه

۲. همیشه رابطه دو طرفه نیاز نیست

۳. با توجه به نیاز برنامه روابط تعریف می شوند

تنظیم پیش فرض روابط

۱. بهینه شده نسبت به نسخه EF 6

۲. تعیین روابط با Navigation Property



Foreign key چگونه پیدا می شود؟

۱. نام کلید اصلی Principal

۲. نام کلاس به همراه کلید اصلی Principal

۳. نام خاصیت و نام کلید Principal



چگونگی تعیین اجباری بودن

۱. Nullable بودن یعنی غیر اجباری

۲. در غیر این صورت اجباری



خاصیت FK ضروری است؟

۱. می‌توان خاصیت جداگانه تعریف نکرد

۲. کمک به توسعه Disconnected

۳. عدم تعریف یعنی Shadow Property

۴. به صورت پیش فرض nullable



چه زمانی پیش فرض کار نمی کند؟



۱. استفاده از Composite Key

۲. روابط یک به یک بدون تعریف دو طرفه

۳. دو خاصیت از جنس یک Entity

۴. نیاز به تعریف Constraint

Data Annotation و روابط

۱. استفاده از ForeignKey

۲. روی Navigation قرار می‌گیرد

۳. حالت برعکس امکان پذیر است.

۴. InverseProperty جهت تعیین انتهای رابطه



Fluent API و روابط

۱. نسبت به EF6 تغییر کرده
۲. قابلیت تعریف رابطه یک به چند
۳. قابلیت تعریف رابطه یک به یک



تنظیم روابط



۱. OnDelete هنگام حذف چه اتفاقی بیوفتد
 ۲. IsRequired تعیین اجباری بودن
 ۳. HasConstraintName نام کلید خارجی
- DB**

IsRequired و کاربرد آن

۱. وجود رابطه را اجباری می‌کند
۲. با Shadow Property ها کاربرد دارد



HasPrincipalKey



۱. کلید ثانویه برای جدول تعیین می‌کند
۲. کدملی یکتا است ولی کلید اصلی نیست
۳. می‌تواند ترکیبی باشد

ویژگی‌های کم کاربرد



۱. `HasConstraintName` نام کلید خارجی

۲. `MetaData` برای اطلاعات اضافه روابط

انواع روش‌های Map کردن کلاس

۱. Owned Types

۲. Table Per Hierarch

۳. Table Splitting



Owned Types



۱. خواص Reference Type که رابطه نیست

۲. تعیین با OwnsOne

۳. در دیتابیس درون یک جدول تعریف می شود

Owned Types



۴. موقع نمونه سازی مقدار نیاز دارند

۵. قابلیت OwnsMany در EF Core
2.2

Table Per Hierarch

۱. برای موجودیت‌ها در روابط ارث‌بری
۲. به صورت پیش‌فرض پشتیبانی می‌شود



Table Per Hierarch

۳. نگهداری تمام اطلاعات در یک جدول

۴. اضافه شدن ستونی جهت تشخیص نوع



Table Splitting



۱. اطلاعات چند Entity در یک جدول
۲. راهی برای پیاده سازی روابط یک به یک
۳. امکان بارگذاری کلی یا جزئی وجود دارد

Configuring advanced features

مدرس: علیرضا ارومند

چه خواهیم آموخت

۱. استفاده از Backing Field در روابط

۲. استفاده از UDF ها

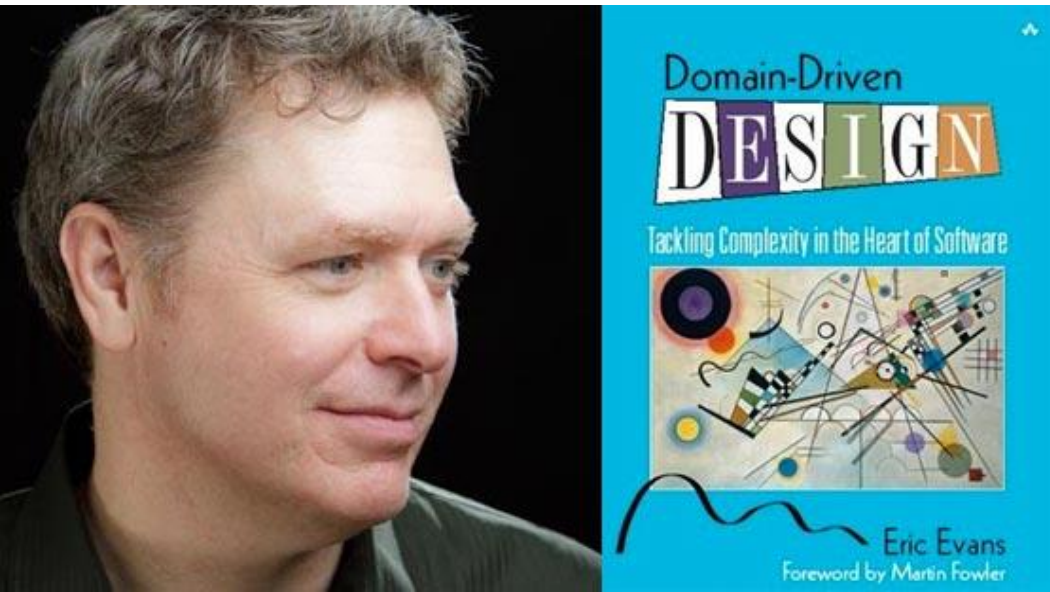
۳. کار با Computed column

۴. SQL Sequence در EF



مدیریت روابط با Backing Field

۱. مناسب برای توسعه DDD
۲. یکی از روش‌های کنترل دسترسی
۳. می‌توان بهبود عملکرد به دست آورد



UDF ها و EF Core

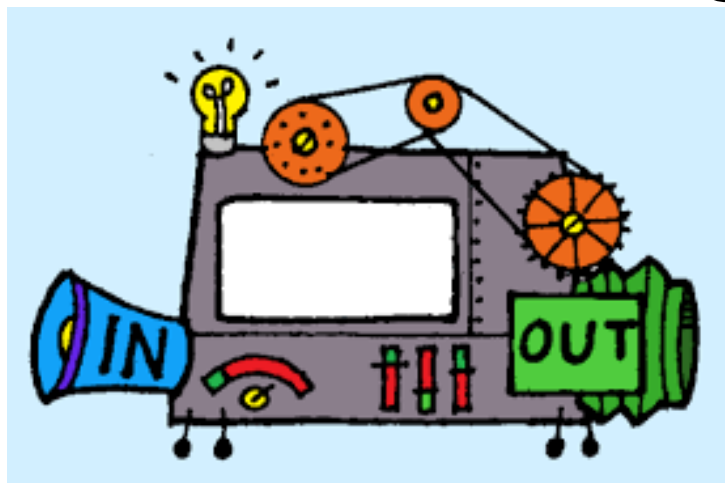
۱. استفاده از DbFunction

۲. مرحله اول تعریف تابع در C#

۳. مرحله دوم برقراری ارتباط با EF

۴. مرحله سوم تعریف UDF

۵. استفاده از UDF



UDF ها و EF Core

۶. امکان تعریف Fluent و Attribute

۷. Attribute محدود به DbContext

۸. ورودی و خروجی و نام توابع یکی است

۹. پیاده سازی ندارد



Computed Column



۱. پشتیبانی در EF 6.x بهینه نبود

۲. فقط خواندنی هستند و مقدارشان محاسباتی

تنظیم مقدار پیش فرض



۱. سه روش قابل بررسی است

۲. تخصیص مقدار پیش فرض به Property

تنظیم مقدار پیش فرض



۳. استفاده از SQL

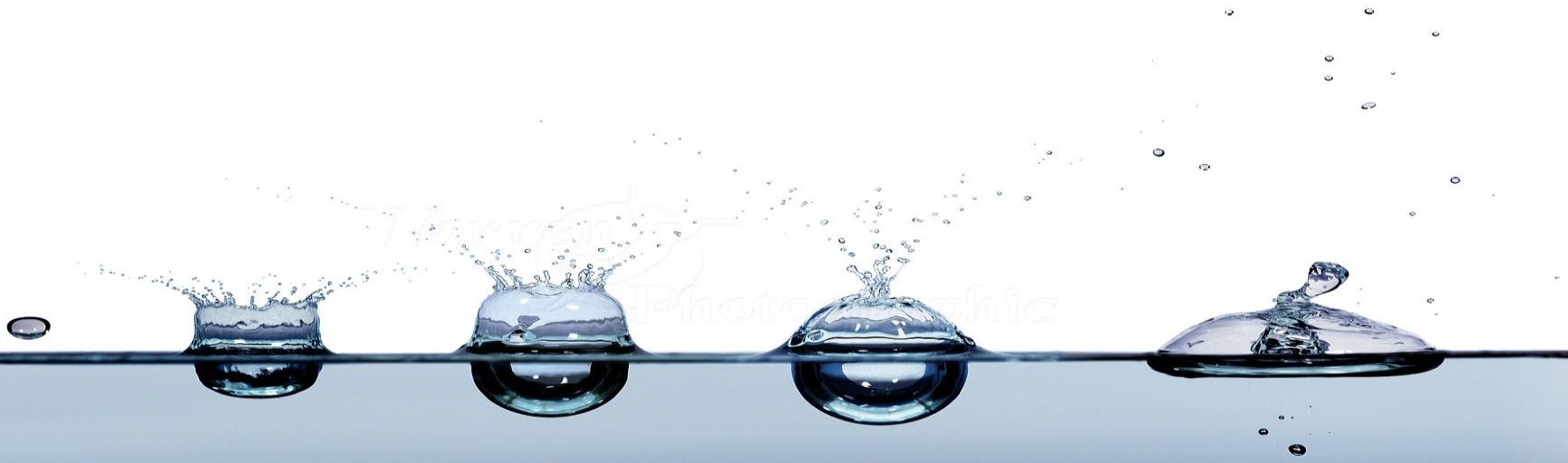
۴. محاسبه مقدار پیش فرض

ایجاد یک توالی

۱. هنگام اهمیت توالی استفاده می‌شود.

۲. تنظیم با `HasSequence`

۳. این قابلیت در `EF6.x` وجود ندارد



تولید مقدار در دیتابیس

۱. امکان تعریف Fluent و Attribute

۲. امکان تعیین زمان تخصیص وجود دارد

۳. امکان فعال سازی و غیرفعال سازی دارد



همزمانی چیست؟



۱. تلاش برای دسترسی همزمان به یک منبع
۲. مشکل در زمان به ثبت اطلاعات رخ می‌دهد
۳. کدام اطلاعات صحیح است؟

راه حل های EF Core

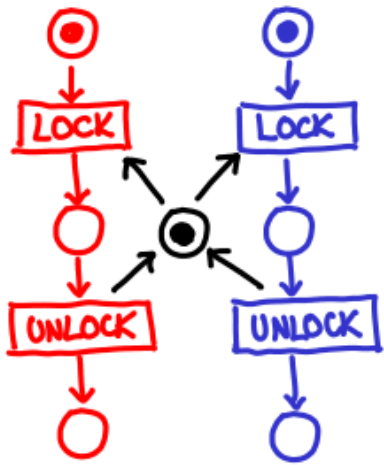
۱. استفاده از Concurrency token

۲. استفاده از TimeStamp

۳. مانده EF 6.x عمل می‌کند

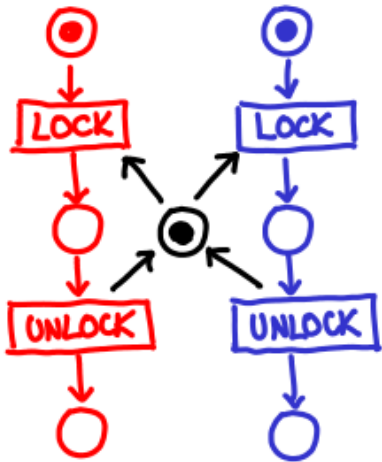


استفاده از Concurrency Token



۱. معرفی یک یا چند خاصیت برای توکن
۲. شرکت در شرط به روزرسانی

استفاده از Concurrency Token



۳. تنظیم با ConcurrencyChek

۴. استفاده از IsConcurrencyToken

چگونگی استفاده از TimeStamp

۱. از امکانات توکار

۲. اجتماع ستون‌ها یکی می‌باشد



مدیریت خطای همزمانی

۱. DbUpdateConcurrencyException

۲. مدیریت خطاها در لایه صحیح



همزمانی در محیط Disconnected

۱. تغییر در فاصله خواندن تا نوشتن

۲. ارائه‌ی وضعیت جدید با خطا

۳. پیاده‌سازی با هر دو روش مدیریت همزمانی



Data Seeding

۱. امکان ثبت داده اولیه

۲. امکان جدید در EF Core 2.1



شبکه‌های اجتماعی نیک آموز

اطلاع رسانی سریع کارگاه‌های نسبتاً رایگان،

کوپن‌های تخفیف، مقالات، فیلم و دوره‌های نیک آموز



Instagram



Telegram

