

در EF 6 امکان تعریف ساده‌تر ایندکس‌ها توسط [data annotations](#) میسر شده‌است. برای مثال:

```
public abstract class BaseEntity
{
    public int Id { get; set; }
}

public class User : BaseEntity
{
    [Index(IsUnique = true)]
    public string EmailAddress { get; set; }
}
```

در اینجا توسط ویژگی Index، خاصیت آدرس ایمیل به صورت منحصر بفرد تعریف شده‌است.

سؤال: چگونه می‌توان شبیه به composite keys، اما نه دقیقاً composite keys، بر روی چند فیلد با هم ایندکس منحصر بفرد تعریف کرد؟

```
public class UserRating : BaseEntity
{
    public VoteSectionType SectionType { set; get; }

    public double RatingValue { get; set; }

    public int SectionId { get; set; }

    [ForeignKey("UserId")]
    public virtual User User { set; get; }
    public int UserId { set; get; }
}
```

در اینجا جدول رای‌های ثبت شده‌ی یک سیستم را مشاهده می‌کنید. می‌خواهیم یک کاربر نتواند بیش از یک رای به یک مطلب خاص بدهد. به عبارتی نیاز است بر روی SectionType (مطلب، اشتراک‌ها، دوره‌ها و ...)، SectionId (شماره مطلب) و UserId (شماره کاربر) یک کلید منحصر بفرد ترکیبی تعریف کرد. ترکیب این سه مورد باید در کل جدول منحصر بفرد باشند (Multiple column indexes).

همچنین نمی‌خواهیم Composite key هم تعریف کنیم. می‌خواهیم Id و Primary key این جدول مانند قبل برقرار باشد. انجام چنین کاری در EF 6.1 به نحو ذیل میسر شده‌است:

```
public class UserRating : BaseEntity
{
    [Index("IX_Single_UserRating", IsUnique = true, Order = 1)] // کلید منحصر بفرد ترکیبی روی سه ستون
    public VoteSectionType SectionType { set; get; }

    public double RatingValue { get; set; }

    [Index("IX_Single_UserRating", IsUnique = true, Order = 2)]
    public int SectionId { get; set; }

    [ForeignKey("UserId")]
    public virtual User User { set; get; }

    [Index("IX_Single_UserRating", IsUnique = true, Order = 3)]
    public int? UserId { set; get; }
}
```

نکته‌ی انجام اینکار، تعریف Indexها با یک نام یکسان صریحاً مشخص شده‌است. در اینجا سه ایندکس تعریف شده‌اند؛ اما نام آن‌ها یکی است و مساوی IX_Single_UserRating قرار داده شده‌است. هر سه مورد نیز IsUnique تعریف شده‌اند و Order آن‌ها نیز

باید مشخص گردد.

خروجی SQL چنین تنظیمی به صورت زیر است:

```
CREATE UNIQUE INDEX [IX_Single_UserRating]
ON [UserRatings] ([SectionType] ASC,[SectionId] ASC,[UserId] ASC);
```

نظرات خوانندگان

نویسنده: رضا

تاریخ: ۱۶:۵۱ ۱۳۹۳/۰۵/۱۸

با سلام :

تفاوت این کار با تعریف در دیتابیس چه می باشد؟
اگر بخواهیم به جای ASC مقدار Desc قرار بگیرد چه کار کنیم؟

نویسنده: وحید نصیری

تاریخ: ۱۸:۲۴ ۱۳۹۳/۰۵/۱۸

EF - جزو خانواده ی ابزارهایی به نام ORMs است. زمانیکه از یک ORM استفاده می کنید و مستقیما SQL نویسی نمی کنید، کدهای شما قابل انتقال می شوند. می توانید به سادگی بانک اطلاعاتی برنامه را عوض کنید بدون اینکه نیازی باشد در کدهای اصلی برنامه تغییری حاصل شود. اهمیت این مساله در اینجا است که نهایتا پروایدر آن بانک اطلاعاتی خاص، بر اساس تعاریف برنامه و ORM مورد استفاده می داند که چگونه باید SQL صحیح و مرتبطی را تولید کند که ممکن است از یک بانک اطلاعاتی به بانک اطلاعاتی دیگری متفاوت باشد.

- فعلا از طریق ویژگی فوق پشتیبانی نمی شود.

نویسنده: داوود

تاریخ: ۱۵:۱۸ ۱۳۹۳/۰۶/۲۰

با سلام

آیا راهی وجود داره که بشه این کار رو تو فایل کانفیگ انجام داد

نویسنده: وحید نصیری

تاریخ: ۱۵:۳۷ ۱۳۹۳/۰۶/۲۰

با روش Fluent

- تک ستونی:

```
modelBuilder.Entity<People>()
    .Property(x => x.Firstname)
    .HasColumnAnnotation("Index", new IndexAnnotation(new IndexAttribute("ix_people_firstname")));
```

- چند ستونی:

```
modelBuilder.Entity<People>()
    .Property(x => x.Firstname)
    .HasColumnAnnotation("Index", new IndexAnnotation(new IndexAttribute("ix_people_fullname", 1)));
modelBuilder.Entity<People>()
    .Property(x => x.Lastname)
    .HasColumnAnnotation("Index", new IndexAnnotation(new IndexAttribute("ix_people_fullname", 2)));
```

- منحصر بفرد:

```
modelBuilder.Entity<People>()
    .Property(x => x.NationalInsuranceNo)
    .HasColumnAnnotation("Index",
        new IndexAnnotation(new IndexAttribute("ix_people_nationalinsurance") {IsUnique = true}));
```

- نوع Clustered

```
modelBuilder.Entity<People>()  
    .Property(x => x.Id)  
    .HasColumnAnnotation("Index",  
        new IndexAnnotation(new IndexAttribute("ix_people_nationalinsurance") { IsClustered = true}));
```

نویسنده: م علیزاده
تاریخ: ۱۳۹۴/۰۸/۰۲ ۱:۳۸

با سلام.
میخواستم بدونم برای ایندکس منحصر به فرد هیچ validation توکاری در EF وجود داره؟ که برای مثال وقتی دو تا رکورد تکراری رو به dbset اضافه میکنیم قبل از savechange عمل validation رو انجام بده، یا اینکه باید به صورت دستی validate کنیم. و اگه آره بهترین روش از نظر بهینه بودن برای این کار چی میتونه باشه؟ ValidateEntity یا Ivalidation object؟ یا...؟
با تشکر از وقتی که میگذارید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۸/۰۲ ۱:۴۶

- خیر. این مورد توسط بانک اطلاعاتی بررسی می‌شود و EF در آن نقشی ندارد و در نهایت در صورت شکست، استثنای مرتبط را از بانک اطلاعاتی به برنامه منتقل می‌کند.
- روش مناسب اعتبار سنجی آن، استفاده از [Remote validation](#) است (مثلا در حین ثبت نام بررسی کند که آیا ایمیل وارد شده در بانک اطلاعاتی موجود است یا خیر).