

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

آموزش پیکر بندی رابطه یک به یک

مدرس: مهندس افشین رفوآ

آموزش پی‌کربندی رابطه یک به یک

پی‌کربندی رابطه یک به یک

ما در اینجا رابطه یک به یک را بین کلاس **Student** و کلاس **StudentAddress** پی‌کربندی می‌کنیم. همان طور که می‌دانید رابطه یک به یک زمانی برقرار می‌شود که کلید اولیه یک جدول تبدیل به **PK** و در جدول دیگری به **FK** (کلید خارجی) تبدیل می‌شود. **StudentId** کلید اولیه جدول **Student** است، بنابراین **StudentId** کلید اولیه باشد و همچنین در جدول **StudentAddress** کلید خارجی می‌باشد تا رابطه یک به یک (یک به صفر یا یک به یک) بین آنها برقرار شود.

پی‌کربندی رابطه یک به صفر یا یک به یک با استفاده از `DataAnnotation`

```
public class Student
{
    public Student() { }
    public int StudentId { get; set; }
    [Required]
    public string StudentName { get; set; }
    [Required]
    public virtual StudentAddress
StudentAddress { get; set; }
}
public class StudentAddress
{
    [Key, ForeignKey("Student")]
    public int StudentId { get; set; }
    public string Address1 { get; set; }
    public string Address2 { get; set; }
    public string City { get; set; }
    public int Zipcode { get; set; }
    public string State { get; set; }
    public string Country { get; set; }
    public virtual Student Student { get;
set; }
```

همان طور که در کلاس **Student** و **StudentAddress** می‌بینید، ما کار خاصی در کلاس **Student** انجام ندادیم زیرا کلاس **StudentId** از قراردادها (**conventions**) پیروی می‌کند بنابراین تبدیل به کلید اولیه (**PK**) می‌شود و اکنون ما از خصوصیت **Key** و **ForeignKey** برای فیلد **StudentId** در کلاس **StudentAddress** برای

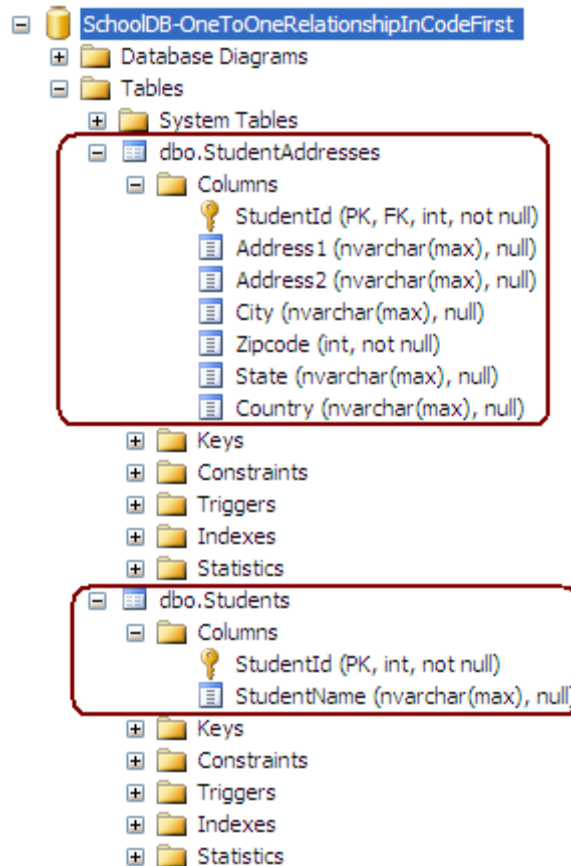
تبدیل آن به کلید اولیه و کلید خارجی استفاده می کنیم. و بنابراین یک رابطه یک به یک بین کلاس های **Student** و **StudentAddress** ایجاد می شود.

پیگردی رابطه یک به یک با استفاده از **Fluent API**

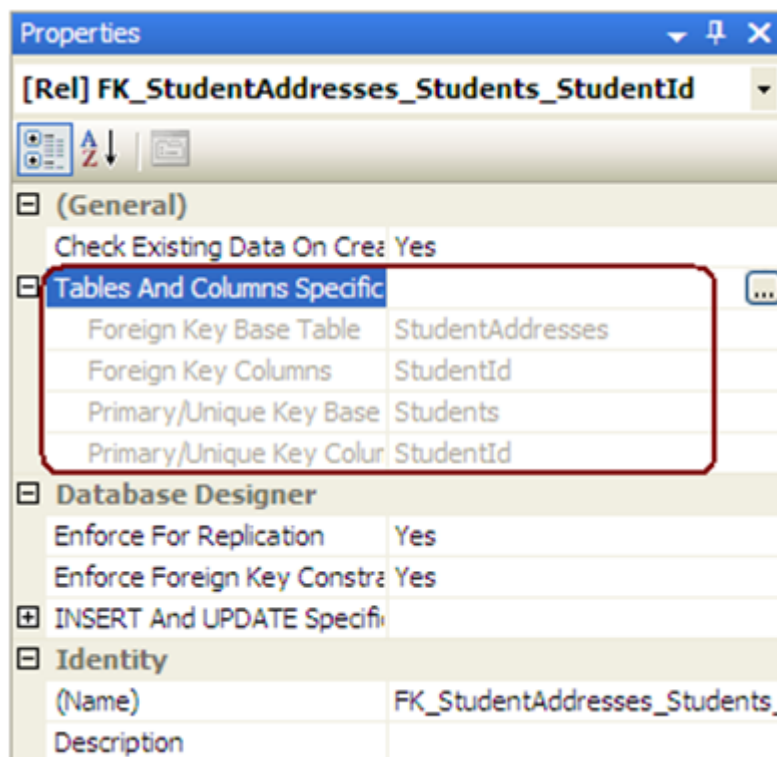
```
protected override void OnModelCreating(DbModelBuilder
modelBuilder)
{
    modelBuilder.Entity()
        .HasKey(e => e.StudentId);
    modelBuilder.Entity()
        .Property(e => e.StudentId)

    .HasDatabaseGeneratedOption(DatabaseGeneratedOption.None);
    modelBuilder.Entity()
        .IsRequired(e => e.Student)
        .WithRequiredDependent(s =>
s.StudentAddress);
    base.OnModelCreating(modelBuilder);
}
```

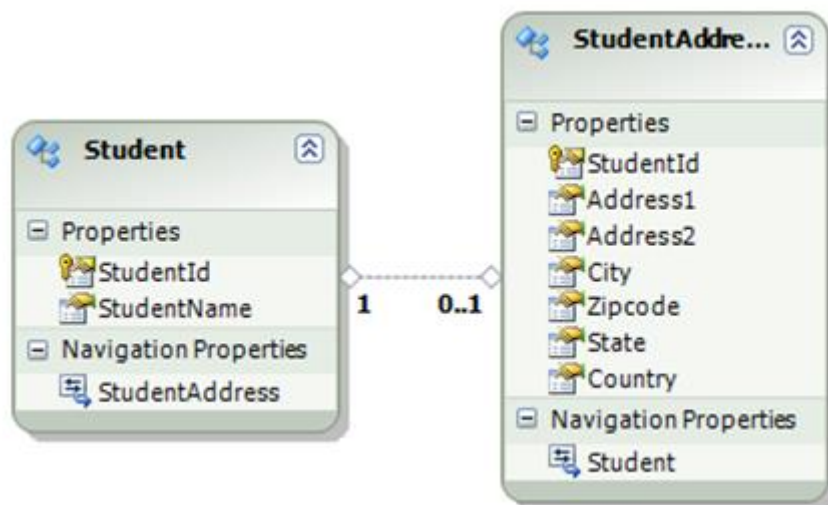
قطعه کد نشان داده شده در بالا پایگاه داده زیر را ایجاد می کند.



شما می توانید رابطه بین کلاس های **Student** و **StudentAddress** را در پایگاه داده را همانطور که در زیر نشان داده شده است را بررسی کنید.



اگر شما یک موجودیت مدل داده ای از پایگاه داده ساخته شده ایجاد کنید، دیگرام آن به شکل زیر نشان داده خواهد شد.



شما نحوه ایجاد رابطه ای یک به چند را در قسمت بعدی یاد خواهید گرفت.

www.tahilidadeh.com