我们已经知道了，Code--First默认的约定，如果你没有包含外键属性在父类中，那么他会为我们创建{Class Name}\_{primary Key}外键。这个InverseProperty特性用在：类之间当有多重关系的时候。

看下下面的代码：

[IMG_256](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

using System;using System.Collections.Generic;using System.Linq;using System.Text;using System.Threading.Tasks;

namespace EF3

{

public class Student

{

public int StudentId { get; set; }

public string StudentName { get; set; }

public Standard CurrentStandard { get; set; }

public Standard PriviousStandard { get; set; }

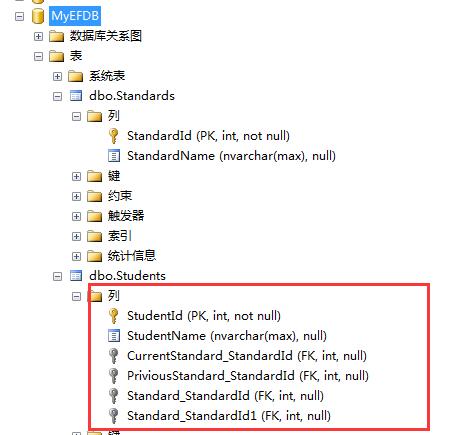
}

}

[IMG_257](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;    namespace EF3  {     public class Standard      {         public int StandardId { get; set; }           public string StandardName { get; set; }           public ICollection<Student> CurrentStudent { get; set; }           public ICollection<Student> PreviousStudent { get; set; }        }  } |

　　上面的代码中，Student实体包含两个Standard类型的导航属性，同样的Standard实体包含两个集合类型的Student导航属性，Code-First创建了4个列为他们之间的关系：



InverseProperty这个特性可以重写这个默认的约定，下面的代码中，我们可以在Standard中使用InverseProperty特性来修正这个问题。

我们先看一下错误的例子：

[IMG_259](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

using System;using System.Collections.Generic;using System.ComponentModel.DataAnnotations.Schema;using System.Linq;using System.Text;

namespace EF3

{

public class Standard

{

public int StandardId { get; set; }

public string StandardName { get; set; }

/// <summary>

/// InverseProperty,反属性特性

/// </summary>

[InverseProperty("one")]

public ICollection<Student> CurrentStudent { get; set; }

/// <summary>

/// InverseProperty,反属性特性

/// </summary>

[InverseProperty("two")]

public ICollection<Student> PreviousStudent { get; set; }

}

}

[IMG_260](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

上面的反属性特性里面我随便输入不存在的字符，然后：



提示这个反属性特性作用的属性CurrnentStudent不合法，这个属性one不是合法的导航属性。

然后我们看下正确的代码：

[IMG_262](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

using System;using System.Collections.Generic;using System.ComponentModel.DataAnnotations.Schema;using System.Linq;using System.Text;

namespace EF3

{

public class Standard

{

public int StandardId { get; set; }

public string StandardName { get; set; }

/// <summary>

/// InverseProperty,反属性特性

/// </summary>

[InverseProperty("CurrentStandard")]

public ICollection<Student> CurrentStudent { get; set; }

/// <summary>

/// InverseProperty,反属性特性

/// </summary>

[InverseProperty("PriviousStandard")]

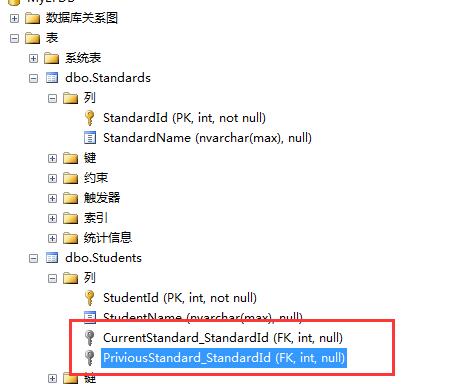
public ICollection<Student> PreviousStudent { get; set; }

}

}

[IMG_263](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

这个代码里面把Student实体中的导航属性的名称放进去。就可以了。



你当然可以使用外键特性来包含外键属性，请看下面的代码：

[IMG_265](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

using System;using System.Collections.Generic;using System.ComponentModel.DataAnnotations.Schema;using System.Linq;using System.Text;using System.Threading.Tasks;

namespace EF3

{

public class Student

{

public int StudentId { get; set; }

public string StudentName { get; set; }

public int ForeignKeyCurrent { get; set; }

public int ForeignKeyPrevious { get; set; }

[ForeignKey("ForeignKeyCurrent")]

public Standard CurrentStandard { get; set; }

[ForeignKey("ForeignKeyPrevious")]

public Standard PriviousStandard { get; set; }

}

}

[IMG_266](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

[IMG_267](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

using System;using System.Collections.Generic;using System.ComponentModel.DataAnnotations.Schema;using System.Linq;using System.Text;

namespace EF3

{

public class Standard

{

public int StandardId { get; set; }

public string StandardName { get; set; }

/// <summary>

/// InverseProperty,反属性特性

/// </summary>

[InverseProperty("CurrentStandard")]

public ICollection<Student> CurrentStudent { get; set; }

/// <summary>

/// InverseProperty,反属性特性

/// </summary>

[InverseProperty("PriviousStandard")]

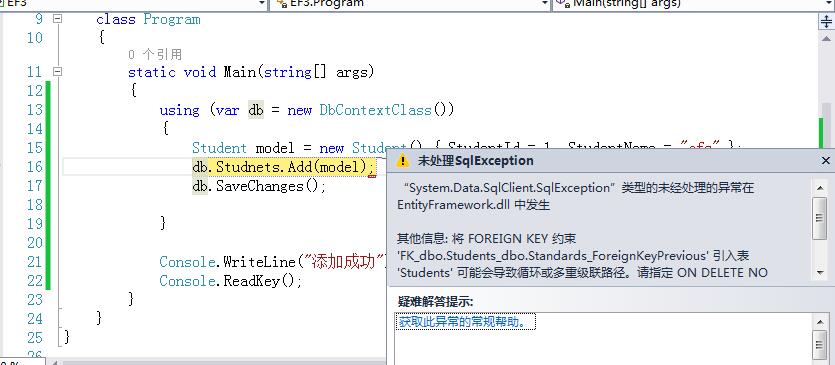
public ICollection<Student> PreviousStudent { get; set; }

}

}

[IMG_268](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

等等，你以为就样就可以了呢？我们运行程序看看就知道了：



果不其然又出错了，看一下具体的信息：

将 FOREIGN KEY 约束 'FK\_dbo.Students\_dbo.Standards\_ForeignKeyPrevious' 引入表 'Students' 可能会导致循环或多重级联路径。请指定 ON DELETE NO ACTION 或 ON UPDATE NO ACTION，或修改其他 FOREIGN KEY 约束。  
无法创建约束。请参阅前面的错误消息。

百度了一下：[http://www.cnblogs.com/chear/archive/2012/11/09/2762145.html](http://www.cnblogs.com/chear/archive/2012/11/09/2762145.html" \t "https://www.cnblogs.com/caofangsheng/p/_blank)

改一下我们的上下文类的代码：

[IMG_270](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

using System;using System.Collections.Generic;using System.Data.Entity;using System.Linq;using System.Text;using System.Threading.Tasks;

namespace EF3

{

public class DbContextClass:DbContext

{

public DbContextClass():base("ConStr") { }

public DbSet<Student> Studnets { get; set; }

public DbSet<Standard> Standards { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

Database.SetInitializer(new DropCreateDatabaseIfModelChanges<DbContextClass>());  
//加上这句代码，就OK了，取消级联删除。（这里其实可以两句代码，只写一句也是可以的。亲测过！！）

modelBuilder.Entity<Standard>().HasMany(t => t.CurrentStudent).WithRequired(p=>p.CurrentStandard).WillCascadeOnDelete(false);

           modelBuilder.Entity<Standard>().HasMany(t => t.PreviousStudents).WithRequired(p => p.PriviousStandard).WillCascadeOnDelete(false);

base.OnModelCreating(modelBuilder);

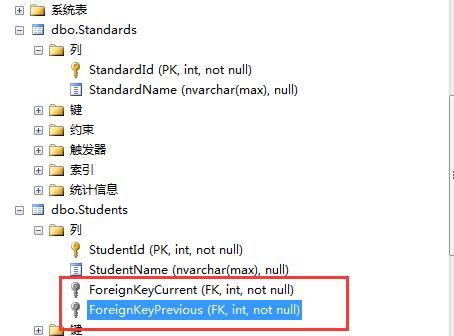
}

}

}

[IMG_271](https://www.cnblogs.com/caofangsheng/p/javascript:void(0);)

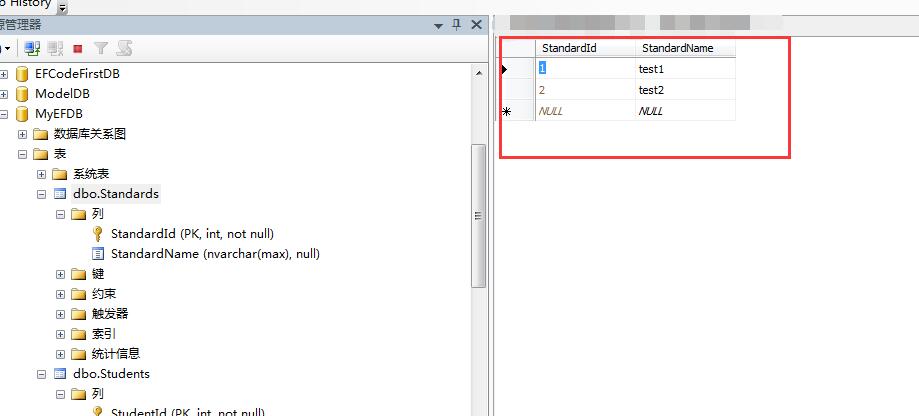
然后再运行程序就可以了。



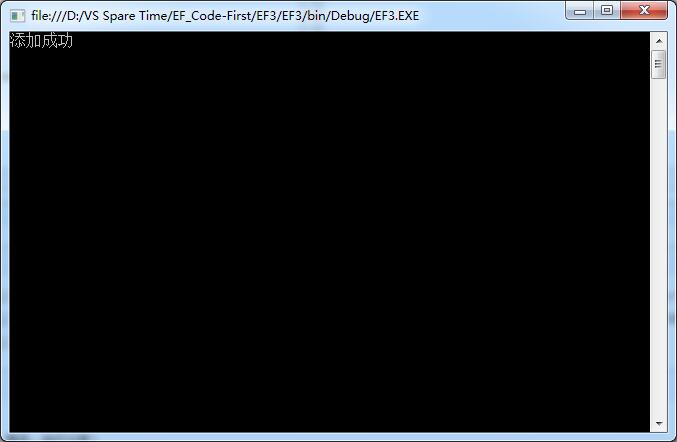
Thus, you can use InverseProperty and ForeignKey attribute for multiple relationships between the same classes.

所以，当有多重关系的时候，你可以使用InverseProperty特性和外键特性。

当我们运行的时候，保存报错，提示主外键冲突什么的，那就是主键表Standard里面没有数据，我们插入两条数据进去，就可以了。



运行之后；



每天学一点，每天积累一天，进步就不止一点点！PS：好记性不如烂笔头，学会总结，学会思考～～～ ----要飞翔，必须靠自己！