Key特性可以被用到类的属性中，Code-First默认约定，创建一个主键，是以属性的名字“Id”，或者是类名+Id来的。

Key特性重写了这个默认的约定，你可以应用Key特性到一个类的属性上面，不管这个属性的名字是什么，你都可以创建一个主键。

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

namespace EF2

{

[Table("StudentInfo")]

public class Student

{

[Key]

[Column(Order=1)]

public int StudentKey1 { get; set; }

[Key]

[Column(Order=2)]

public int StudentKey2 { get; set; }

[Column("Name",TypeName="ntext")]

[MaxLength(20)]

public string StudentName { get; set; }

[NotMapped()]

public int? Age { get; set; }

public int StdId { get; set; }

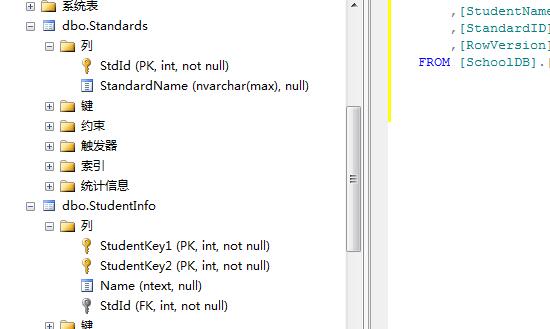
[ForeignKey("StdId")]

public virtual Standard Standard { get; set; }

}

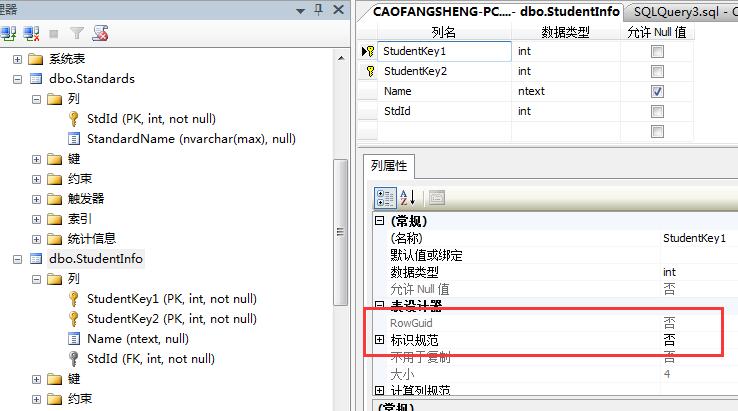
}

运行程序之后，看下数据库：



请注意：单个主键，EF Code-First为我们创建的：主键是自动增长的，而复合主键，得到的主键不是自动增长的。

当然，key 属性不仅仅可以引用到int类型的属性上，还可以应用到字符串，日期类型等等。。



好了，这就是数据注解特性的Key特性了。