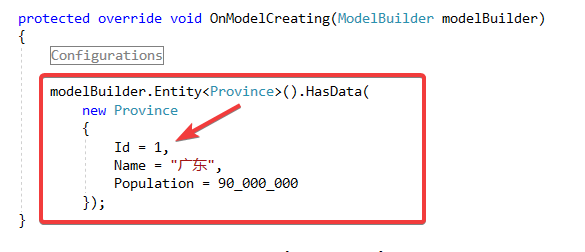
**在DBContext中配置种子数据**

在DBContext的OnModelCreating方法里使用**HasData()**方法:



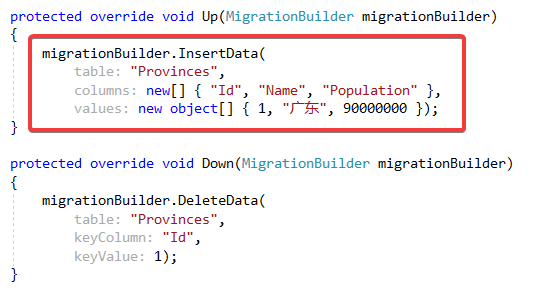
这里我添加了一个省份的种子数据，并**写上了主键Id的值**。

数据库该表的主键Id是int自增的。

**生成的迁移类**

命令：Add-Migration Xxx

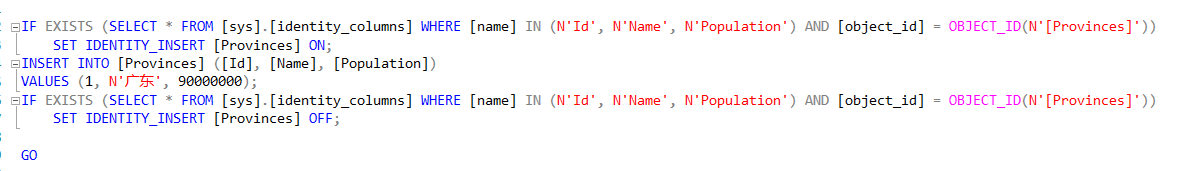
看一下生成的迁移类的内容：



**生成的SQL脚本**

命令：Script-Migration

这是里面关于插入数据的部分：

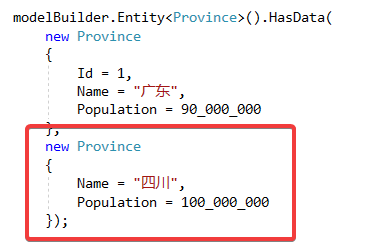


**迁移到数据库**

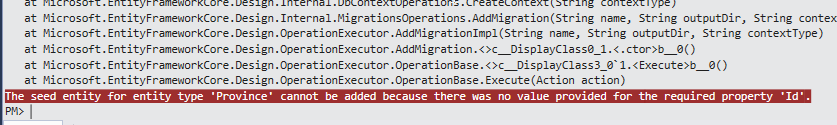
命令：Update-Database -Verbose

**种子数据的主键必须有值**

我再添加一个没有主键Id值的种子数据：



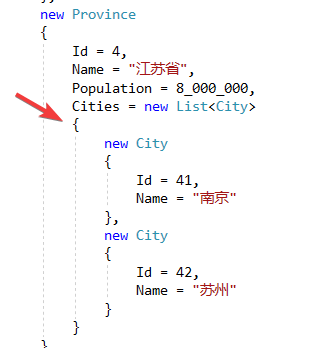
 然后Add-Migration，看看会发生什么：



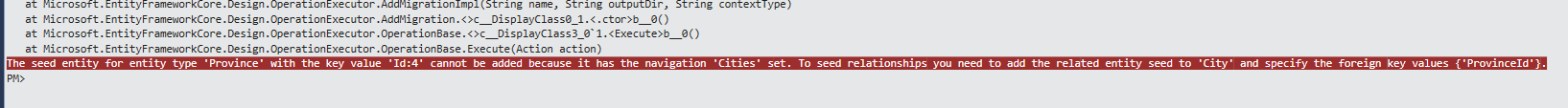
报错了，所以**主键值是必填**的。

**添加关联种子数据**

如下关联种子数据的添加错误示例

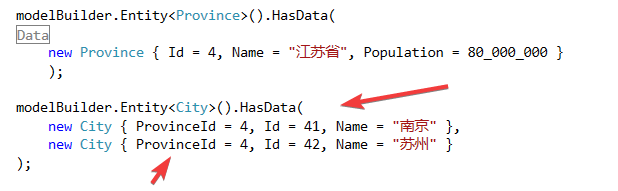


 然后Add-Migration



这样做不行。我**必须单独添加City的种子数据，并且设置好外键**。

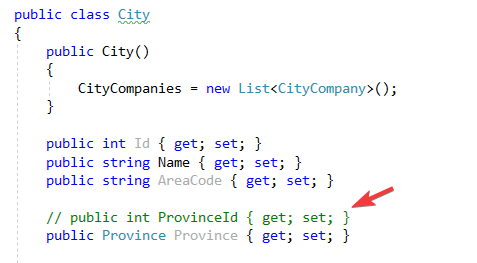
所以正确的做法是：



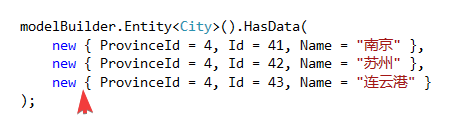
**如果无法在Model里设置主键/外键**

如果关联键是protected类型，或没有关联键，则使用**匿名类**。

我把City Model里的外键去掉（导航属性仍然保留，和Province的主从关系依然存在）：



 可以这样添加种子数据



**值类型的种子数据**

如果某个类型为OwnsOne值类型则如下

builder.OwnsOne (e => e.Creator).HasData (

    new {

        PageDataId = 1,

        EditorId = 1,

        Name = "超级管理员",

        HeadSculpture = "",

        Time = DateTime.Now,

    });

public class EntityEdit {

protected EntityEdit () { }

    public EntityEdit (int editorId, string name, string headSculpture) {

        EditorId = editorId;

        Name = name;

        HeadSculpture = headSculpture;

        Time = DateTime.Now;

}

public int EditorId { get; protected set; }

public string Name { get; protected set; }

public string HeadSculpture { get; protected set; }

    public DateTime Time { get; protected set; }

}