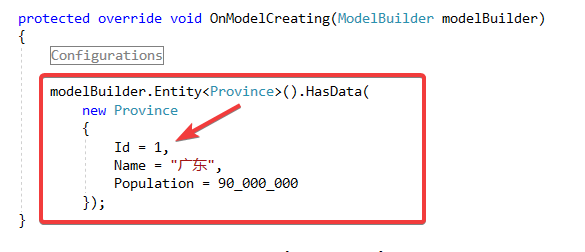
# **添加第一个种子数据**

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



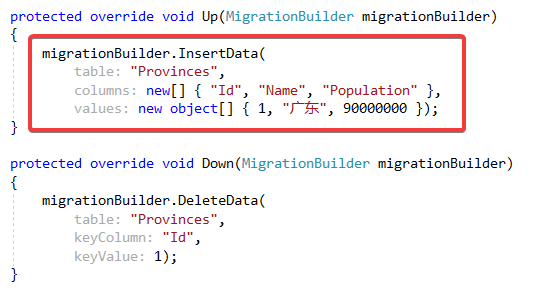
这里我添加了一个省份的种子数据，并**写上了主键Id的值**。

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

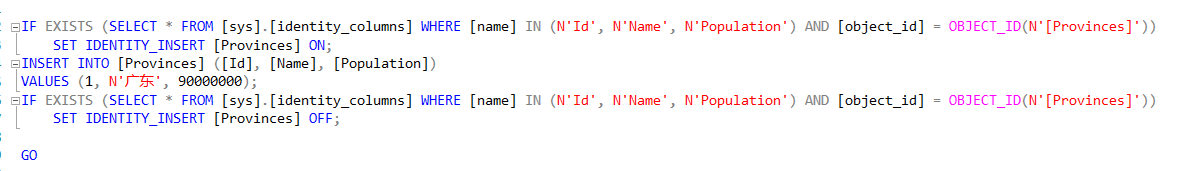
看一下生成的迁移类的内容：



## **生成的SQL脚本**

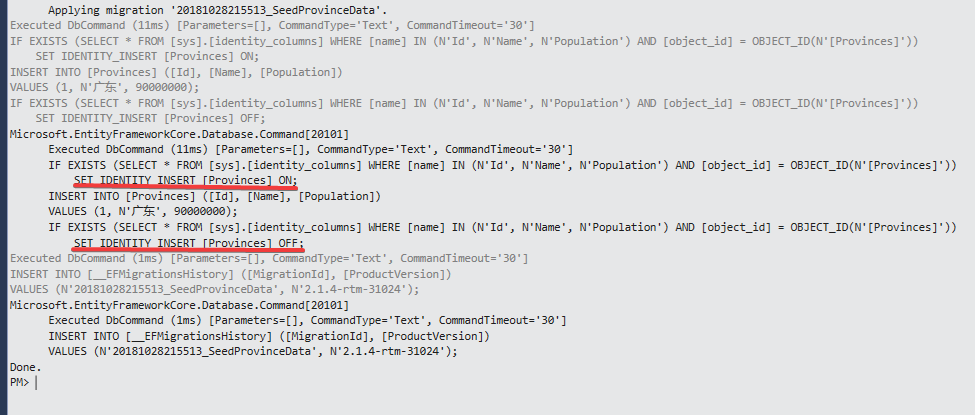
命令：Script-Migration

这是里面关于插入数据的部分：



## **迁移到数据库**

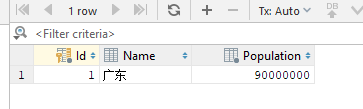
命令：Update-Database -Verbose



结果是成功的。

看红线那两句话，EFCore在执行的过程中临时更改了设置，可以插入主键的值，然后又禁用了插入主键。

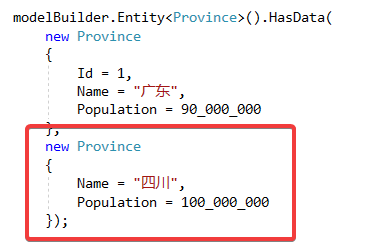
## **数据库里面的数据**



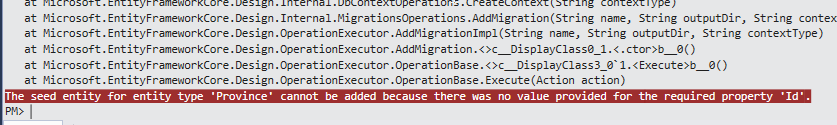
虽然曾经存在过Id为1的数据（然后被删除了），但是Id为1的种子数据仍然可以插入进去。

# **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：



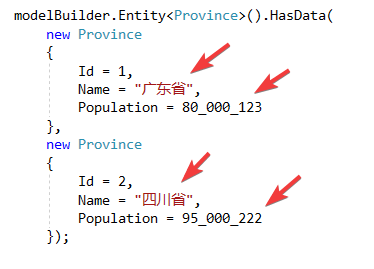
然后Add-Migration，看看会发生什么：



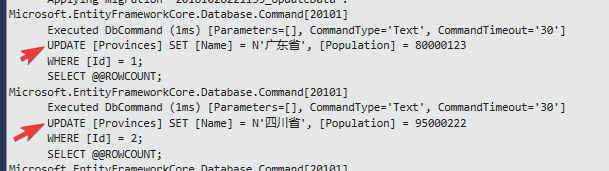
报错了，所以**主键值是必填**的。

## **更改现有的种子数据**

我在HasData方法里更改了现有的种子数据，但是主键的值并没有改：



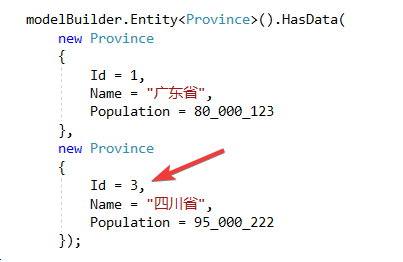
执行Update-Database时的SQL语句：



可以看到是根据主键对数据库里面的数据进行Update动作。

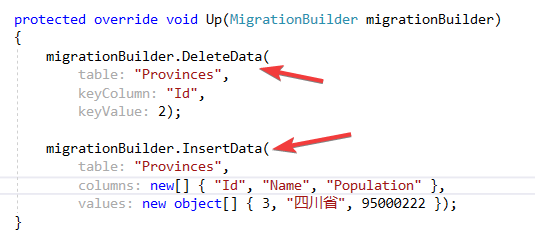
其结果也和我想的一样，就是更新了现有的数据：

## **如果我把HasData里种子数据的主键值修改了**



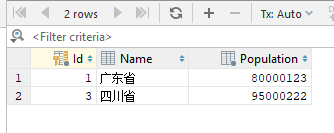
我把四川的主键从2改为3。

看下生成的迁移文件：



先删除了之前添加的Id为2的种子数据，然后把插入了一笔Id为3的数据。

数据库里：



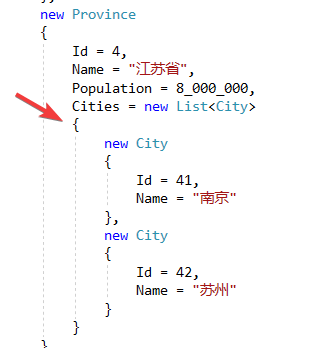
## **种子数据为什么要指定主键的值？**

**因为在团队开发时，这样可以确保不同的开发人员、电脑、服务器上，在同一个迁移版本具有相同的种子数据**。

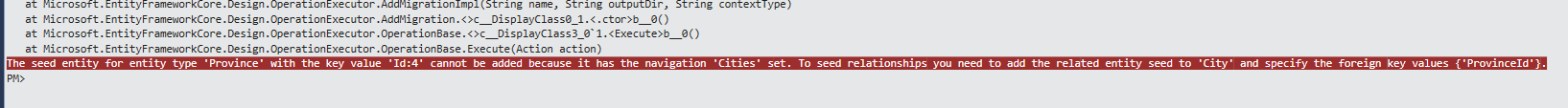
# **添加关联种子数据**

Province和City是一对多的关系，也就是说一个Province可以有多个City，而且它们之间有导航属性。

下面看看一次性添加Province和City是否可以行，我直接在HasData方法里这样写：

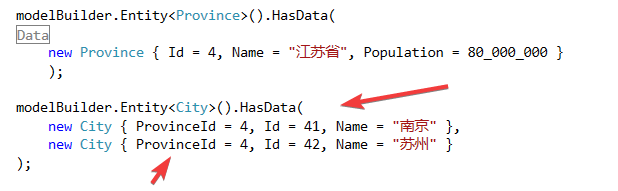


然后Add-Migration

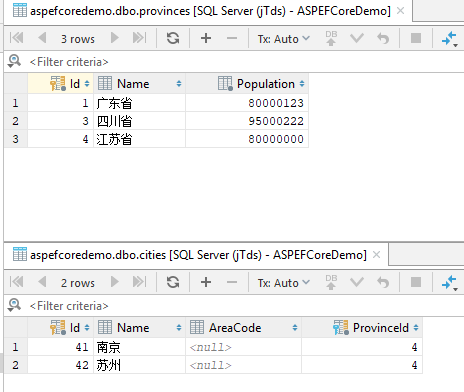


这样做不行。我**必须单独添加City的种子数据，并且设置好外键**。

所以正确的做法是：



这次Add-Migration没有报错，迁移也成功了，看一下最后的数据：



OK

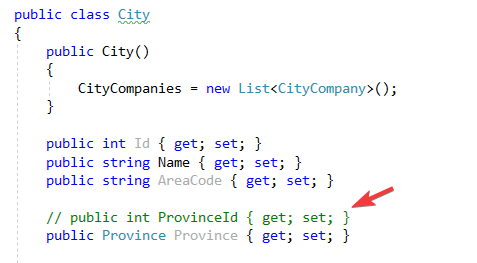
# **如果无法在Model里设置主键/外键**

有时，我们在主从关系的Model里不明确定义外键；有时候我们Model的主键是private set的；

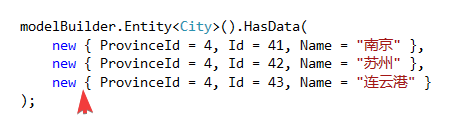
这时我们就无法在HasData里设置主键/外键的值了，那么如何来添加种子数据呢？

答案就是使用**匿名类**。

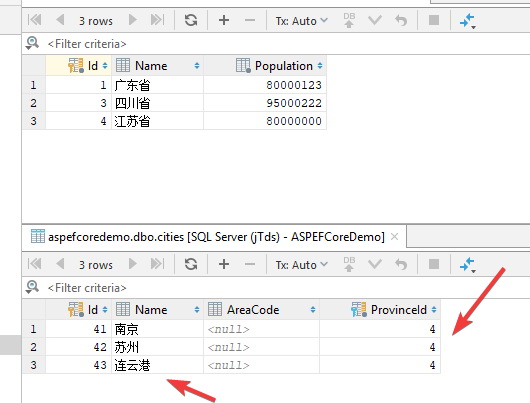
我把City Model里的外键去掉（导航属性仍然保留，和Province的主从关系依然存在）：



然后就可以这样添加种子数据：

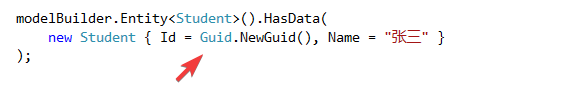


迁移后的数据：

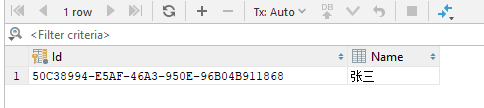


结果仍然如预期一样。

## **如果主键是Guid类型呢？**



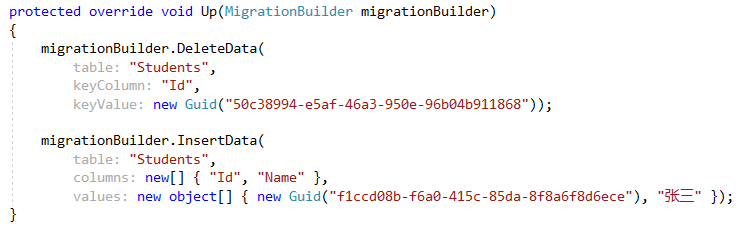
看下数据：



貌似没问题。

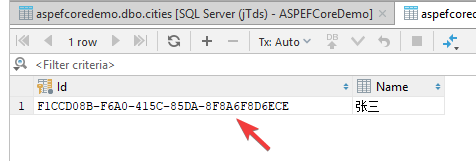
如果我不修改这个种子数据，再执行一次迁移呢？

看一下这时的迁移文件：

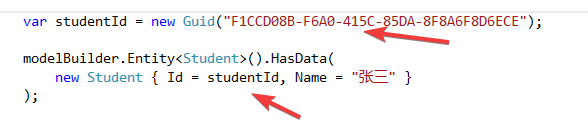


删除原来的数据，再插入一个新的数据。。

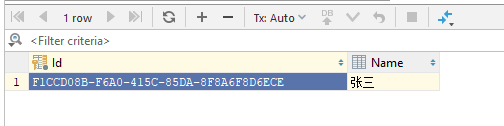
数据库里也是这样的：



所以最好的办法是**把Guid的值放在一个变量里**：



然后再操作一遍：



这样就不会出现“把原有数据删掉，再重新插入”这种操作了。