



Наследование в Entity Framework

Подход ТРН

Последнее обновление: 31.10.2015



При использовании подхода ТРН (Table Per Hierarchy / Таблица на одну иерархию классов) для одной иерархии классов используется одна таблица. Данные базовых и производных классов сохраняются в одну таблицу, а для их отличия создается специальный столбец.

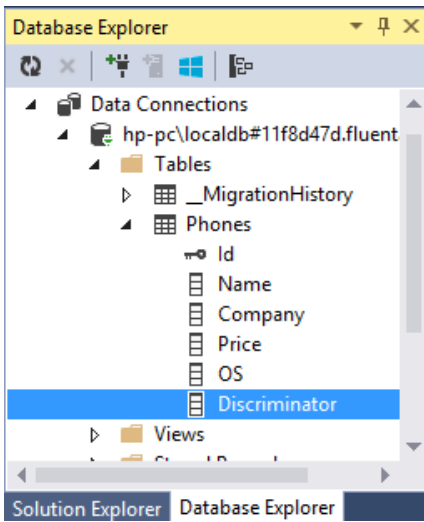
Например, создадим мини-иерархию из двух классов:

```
public class Phone
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Company { get; set; }
    public int Price { get; set; }
}

public class Smartphone : Phone
{
    public string OS { get; set; }
}

class MobileContext : DbContext
{
    public MobileContext() : base("DefaultConnection")
    { }
    public DbSet<Phone> Phones { get; set; }
    public DbSet<Smartphone> Smarts { get; set; }
}
```

Здесь класс Smartphone наследуется от Phone, определяя одно свойство в дополнение к унаследованным. И при работе будет создана такая таблица:



Кроме всех свойств классов Phone и Smartphone здесь также появляется еще один столбец - Discriminator. Он имеет тип nvarchar (то есть строка) и имеет длину в 128 символов. Данный столбец и будет определять относится строка к типу Phone или Smartphone.

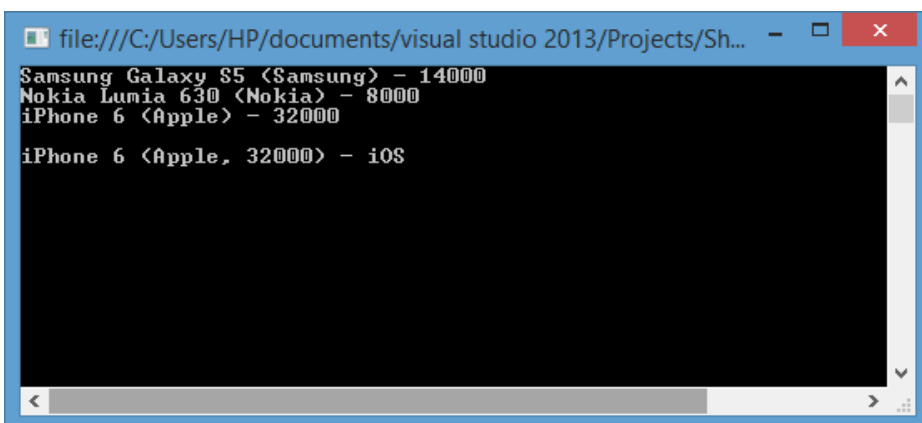
Теперь используем в программе:

```
using(MobileContext db = new MobileContext())
{
    db.Phones.Add(new Phone {Name = "Samsung Galaxy S5", Company = "Samsung", Price = 14000 });
    db.Phones.Add(new Phone {Name = "Nokia Lumia 630", Company = "Nokia", Price = 8000 });

    Smartphone s1 = new Smartphone { Name = "iPhone 6", Company = "Apple", Price = 32000, OS = "iOS" };
    db.Smarts.Add(s1);
    db.SaveChanges();

    foreach (Phone p in db.Phones)
        Console.WriteLine("{0} ({1}) - {2}", p.Name, p.Company, p.Price);
    Console.WriteLine();
    foreach (Smartphone p in db.Smarts)
        Console.WriteLine("{0} ({1}, {2}) - {3}", p.Name, p.Company, p.Price, p.OS);
}
```

Также обратите внимание на вывод консоли:



Так как объект Smartphone также является и объектом Phone, то он также извлекается через db.Phones.

[Назад](#) [Содержание](#) [Вперед](#)



[Вконтакте](#) | [Twitter](#) | [Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2012-2017. Все права защищены.