



Работа с комплексными типами

Последнее обновление: 31.10.2015



Каждый тип сопоставляется Entity Framework с отдельной таблицей. Однако бывают случаи, когда определенный класс существует не сам по себе, а несет некоторую дополнительную информацию по отношению к другой главной модели. Например:

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

    public PhoneInfo Info { get; set; }

    public Phone()
    {
        Info = new PhoneInfo { Price = 300 };
    }
}
```

Класс PhoneInfo содержит некоторую дополнительную информацию по отношению к модели Phone, а модель Phone создает объект PhoneInfo с данными по умолчанию. Таким типом еще называют комплексными.

Комплексные типы имеют ряд ограничений:

- Они не имеют ключей
- Они могут содержать свойства только примитивных типов
- При использовании моделями модель может использовать только один объект типа, но не коллекцию (то есть коллекцию PhoneInfo модель Phone содержать не может)

Чтобы обозначить связь между двумя типами, необходимо использовать атрибут **ComplexType**. Он позволяет создать из разных классов одну сущность.

Например:

```
[ComplexType]
public class PhoneInfo
{
    //.....
}
```

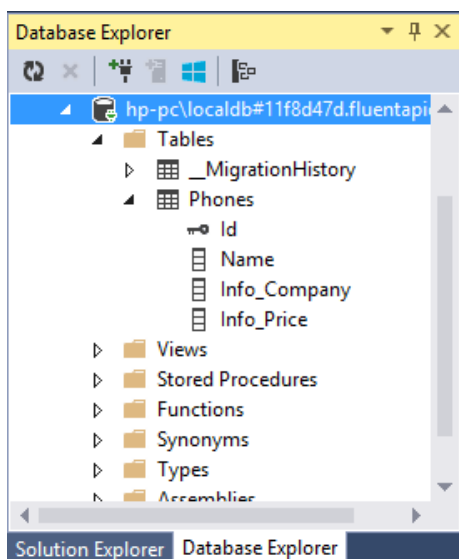
Класс Phone остается без изменений. Тогда мы можем использовать данные типы следующим образом:

```

using(MobileContext db = new MobileContext())
{
    db.Phones.Add(new Phone
    {
        Name = "Samsung Galaxy S5",
        Info = new PhoneInfo { Company = "Samsung", Price = 17000 }
    });
    db.Phones.Add(new Phone
    {
        Name = "Nokia Lumia 930",
        Info = new PhoneInfo { Company = "Nokia", Price = 15000 }
    });
    db.SaveChanges();
    foreach (Phone p in db.Phones)
        Console.WriteLine("{0} - {1}", p.Name, p.Info.Price);
}

```

Тогда при создании таблицы она будет объединять столбцы для обеих моделей, представляя как бы единую сущность:



Альтернативный вариант применению атрибута представляет использование метода **ComplexType** в FluentAPI:

```

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

    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        modelBuilder.ComplexType<PhoneInfo>();
        base.OnModelCreating(modelBuilder);
    }
}

```

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



Стилизатор изображений
Ашманова. ✓

ashmanov.net

Яндекс.Директ

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

Контакты для связи: metanit22@mail.ru

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