



Разделение сущности на несколько таблиц

Последнее обновление: 31.10.2015



В предыдущей теме два класса объединялись в одну таблицу. Но мы также можем сделать и обратное действие - сохранить разные свойства одного класса в разных таблицах. Например, нам потребовалось важную информацию о модели сохранить в одной таблице, а вспомогательную информацию - в другой.

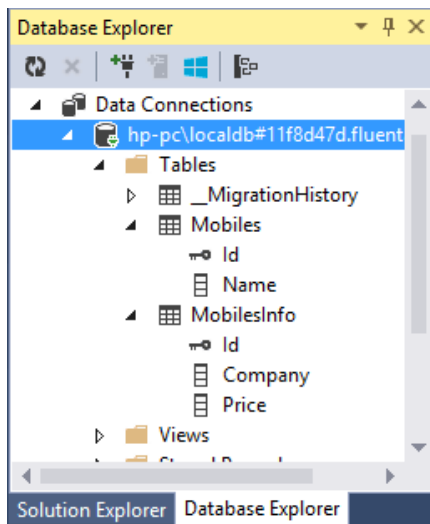
```
public class Phone
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Company { get; set; }
    public int Price { get; set; }
}

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

    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        modelBuilder.Entity<Phone>().Map(m =>
        {
            m.Properties(d => new { d.Name, d.Id });
            m.ToTable("Mobiles");
        }).Map(m =>
        {
            m.Properties(d => new { d.Company, d.Price });
            m.ToTable("MobilesInfo");
        }); ;
        base.OnModelCreating(modelBuilder);
    }
}
```

С помощью метода Map осуществляется сопоставление отдельных свойств с таблицами. Так, данные для свойства Name будут храниться в таблице *Mobiles*, а данные для свойств Company и Price - в таблице *MobilesInfo*. Id мы можем опустить, так как данное поле будет указываться для обеих таблиц для их связи.

В итоге сформируются две таблицы:



Работа с моделью будет проходить аналогично стандартным моделям:

```
using(MobileContext db = new MobileContext())
{
    Phone p1 = new Phone { Id = 1, Name = "Samsung Galaxy S5", Company = "Samsung", Price = 14000 };
    Phone p2 = new Phone { Id = 2, Name = "Nokia Lumia 630", Company = "Nokia", Price = 8000};

    db.Phones.Add(p1);
    db.Phones.Add(p2);
    db.SaveChanges();

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

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

