



Связь один ко многим

Последнее обновление: 17.12.2015



Связь один-ко-многим реализуется, если одна модель хранит ссылку на один объект другой модели, а вторая модель может ссылаться на коллекцию объектов первой модели. Например, в одной команде играет несколько игроков:

```
public class Player
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Position { get; set; }
    public int Age { get; set; }

    public int? TeamId { get; set; }
    public Team Team { get; set; }
}

public class Team
{
    public int Id { get; set; }
    public string Name { get; set; } // название команды

    public ICollection<Player> Players { get; set; }
    public Team()
    {
        Players = new List<Player>();
    }
}

public class SoccerContext : DbContext
{
    public SoccerContext() : base("SoccerContext")
    {}

    public DbSet<Player> Players { get; set; }
    public DbSet<Team> Teams { get; set; }
}
```

Добавление и вывод:

```
using(SoccerContext db = new SoccerContext())
{
    // создание и добавление моделей
    Team t1 = new Team { Name = "Барселона" };
    Team t2 = new Team { Name = "Реал Мадрид" };
    db.Teams.Add(t1);
    db.Teams.Add(t2);
    db.SaveChanges();
    Player p11 = new Player {Name = "Роналду", Age = 31, Position = "Нападающий", Team=t2 };
```

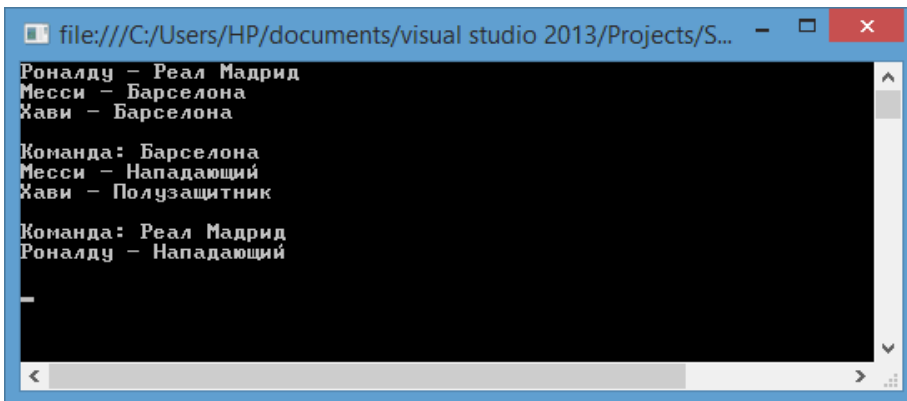
```

Player p12 = new Player { Name = "Месси", Age = 28, Position = "Нападающий", Team=t1 };
Player p13 = new Player { Name = "Хави", Age = 34, Position = "Полузащитник", Team=t1 };
db.Players.AddRange(new List<Player>{p11, p12,p13});
db.SaveChanges();

// вывод
foreach(Player p1 in db.Players.Include(p=>p.Team))
    Console.WriteLine("{0} - {1}", p1.Name, p1.Team!=null?p1.Team.Name:"");
Console.WriteLine();
foreach(Team t in db.Teams.Include(t=>t.Players))
{
    Console.WriteLine("Команда: {0}", t.Name);
    foreach(Player p1 in t.Players)
    {
        Console.WriteLine("{0} - {1}", p1.Name, p1.Position);
    }
    Console.WriteLine();
}
}

```

Результат вывода:



Редактирование:

```

//редактирование
t2.Name = "Реал М."; // изменим название
db.Entry(t2).State = EntityState.Modified;
// переведем игрока из одной команды в другую
p13.Team = t2;
db.Entry(p13).State = EntityState.Modified;
db.SaveChanges();

```

При удалении объектов, связанных отношением "один-ко-многим" нам надо учитывать то, что по умолчанию даже если внешний ключ допускает значение null (как в данном случае свойство TeamId в классе Player), мы не сможем просто так удалить одну модель, если она имеет ссылки на другую модель. Например, удаление команды в данном случае выльется в ошибку, если какой-то объект Player имеет ссылку на эту команду. Что делать в этом случае? В этом случае нам надо установить для внешнего ключа TeamId в таблице игроков ограничение ON DELETE SET NULL. Данное ограничение позволит при удалении связанного объекта устанавливать для внешнего ключа значение null.

Установку ограничения можно сделать вручную, изменим схему базы данных, либо программно. Рассмотрим, как это сделать программно. Для этого перед удалением нам надо выполнить соответствующую команду SQL, чтобы добавить ограничение:

```

db.Database.ExecuteSqlCommand("ALTER TABLE dbo.Players ADD CONSTRAINT Players_Teams FOREIGN KEY (TeamId) REFERENCES dbo.Teams (Id) ON DELETE SET NULL");

```

Здесь добавляется ограничение "Players_Teams" в таблицу игроков, которая называется, как правило, dbo.Players или просто Players. Это ограничение указывает, что для внешнего ключа TeamId из таблицы Players, который связан со столбцом Id из таблицы dbo.Teams, устанавливается правило "ON DELETE SET NULL", то есть установка null по удалению.

И после этого мы можем выполнить удаление:

```
//удаление игрока
Player p1_toDelete = db.Players.First(p => p.Name == "Роналду");
db.Players.Remove(p1_toDelete);
// удаление команды
Team t_toDelete = db.Teams.First();
db.Teams.Remove(t_toDelete);
db.SaveChanges();
```

При удалении команды свойство Team у объектов Player получает значение null.

[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.