**废话（可以不看）**

说到旧项目就很头疼，你说表A有主键Id，但逻辑上却是用字段Code作为主键，表B关联到表A，但表B却是通过自动ACode关联到表A的Code，而不是关联到表A的主键，这样做肯定存在性能问题，具体就不多数，还有一点就是关系好乱

**配置**

实体如下：

// 实体1

public class Entity1

{

    [Key]

    public int Id { get; set; }

    // Entity1 主键

    public string Code { get; set; }

    // 导航属性

    public ICollection<Entity2> Entity2s { get; set; }

}

// 实体2

public class Entity2

{

    [Key]

    public int Id { get; set; }

    // 关联到 Entity1.Code

    public string Entity1Code { get; set; }

    // 导航属性

    public Entity1 Entity1 { get; set; }

}

DB配置如下：

public class MyContext : DbContext

{

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

    {

        optionsBuilder.UseSqlServer("Data Source=(localdb)\\ProjectsV13;Initial Catalog=Test;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

    }

    protected override void OnModelCreating(ModelBuilder modelBuilder) {

        modelBuilder.Entity<Entity1>()

            // 配置 1 对 多

            .HasMany(e => e.Entity2s).WithOne(e => e.Entity1)

            // 配置 Entity2 关联到 Entity1 的外键

            .HasForeignKey(e => e.Entity1Code)

            // 配置 Entity1 在这个关联关系中使用的主键

            .HasPrincipalKey(e => e.Code);

    }

    public DbSet<Entity1> Entity1s { get; set; }

}

**示例1**

添加：

static void Main(string[] args)

{

    MyContext myContext = new MyContext();

    // 添加 Entity1 和其关联的 Entity2

    myContext.Entity1s.Add(new Entity1()

    {

        Code = "Entity1\_1",

        Entity2s = new List<Entity2>() {

            new Entity2(),

            new Entity2()

        }

    });

    myContext.SaveChanges();

}

数据库插入情况：

Entity1

|  |  |
| --- | --- |
| 1 | Entity1\_1 |

Entity2

|  |  |
| --- | --- |
| 1 | Entity1\_1 |
| 2 | Entity1\_1 |

**示例2**

查看：

static void Main(string[] args)

{

    MyContext myContext = new MyContext();

    var entity1\_1 = myContext.Entity1s.Include(e => e.Entity2s).FirstOrDefault(e => e.Code == "Entity1\_1");

    entity1\_1.Entity2s.ToList().ForEach(item =>

    {

        Console.WriteLine(item.Id);

        Console.WriteLine(item.Entity1Code);

    });

}

输出：

1

Entity1\_1

2

Entity1\_1