

EF Code First - Many-to-many

Bei der Arbeit mit HasData und many-to-many Beziehungen muss man folgende Schritte beachten:

Navigation Property als int

Man muss bei den NavigationProperties die Property mit „int“ zusätzlich angeben, also z.B. InstructorId in Klasse Course:

```
public class Course
{
    public int CourseId { get; set; }
    public string? CourseName { get; set; }
    public int InstructorId { get; set; } //<-----
    public Instructor? Instructor { get; set; }
    public List<Student> Students { get; set; } = new();
}
```

Zur Erinnerung, Kap. 5.2.3 im Tutorial: „Aufpassen muss man bei Foreign Keys. Man muss neben der Navigation Property auch noch die Id als Foreign Key angeben.“

HasData

Das many-to-many muss man dann etwas ungewohnt mit String für die Tabelle und anonymen Objekt für die Keys angeben (siehe letzte Zeile).

```
public static void Seed(this ModelBuilder modelBuilder)
{
    Console.WriteLine("DbSeederExtension::Seed");

    Console.WriteLine("Student");
    var student1 = new Student
    {
        StudentId = 1,
        StudentName = "Sandro",
        StudentAddress = "xxxxxxx",
    };
    modelBuilder.Entity<Student>().HasData(student1);

    Console.WriteLine("Instructor");
    var instructor2 = new Instructor
    {
        InstructorId = 2,
        Faculty = "HTL Grieskirchen",
        Name = "Wiederkehr"
    };
    modelBuilder.Entity<Instructor>().HasData(instructor2);

    Console.WriteLine("Course");
    var course1 = new Course
    {
        CourseId = 1,
        CourseName = "English",
        InstructorId = 2
    };
    modelBuilder.Entity<Course>().HasData(course1);

    Console.WriteLine("CourseStudent");
    modelBuilder.Entity("CourseStudent").HasData(new //<-----
    {
        CoursesCourseId = 1,
        StudentsStudentId = 1
    });
}
```

Zwischentabelle

Aus diesen Gründen ist zu empfehlen, die von der Datenbank automatisch erzeugte Zwischentabelle im Datenmodell selbst als Klasse zu notieren.