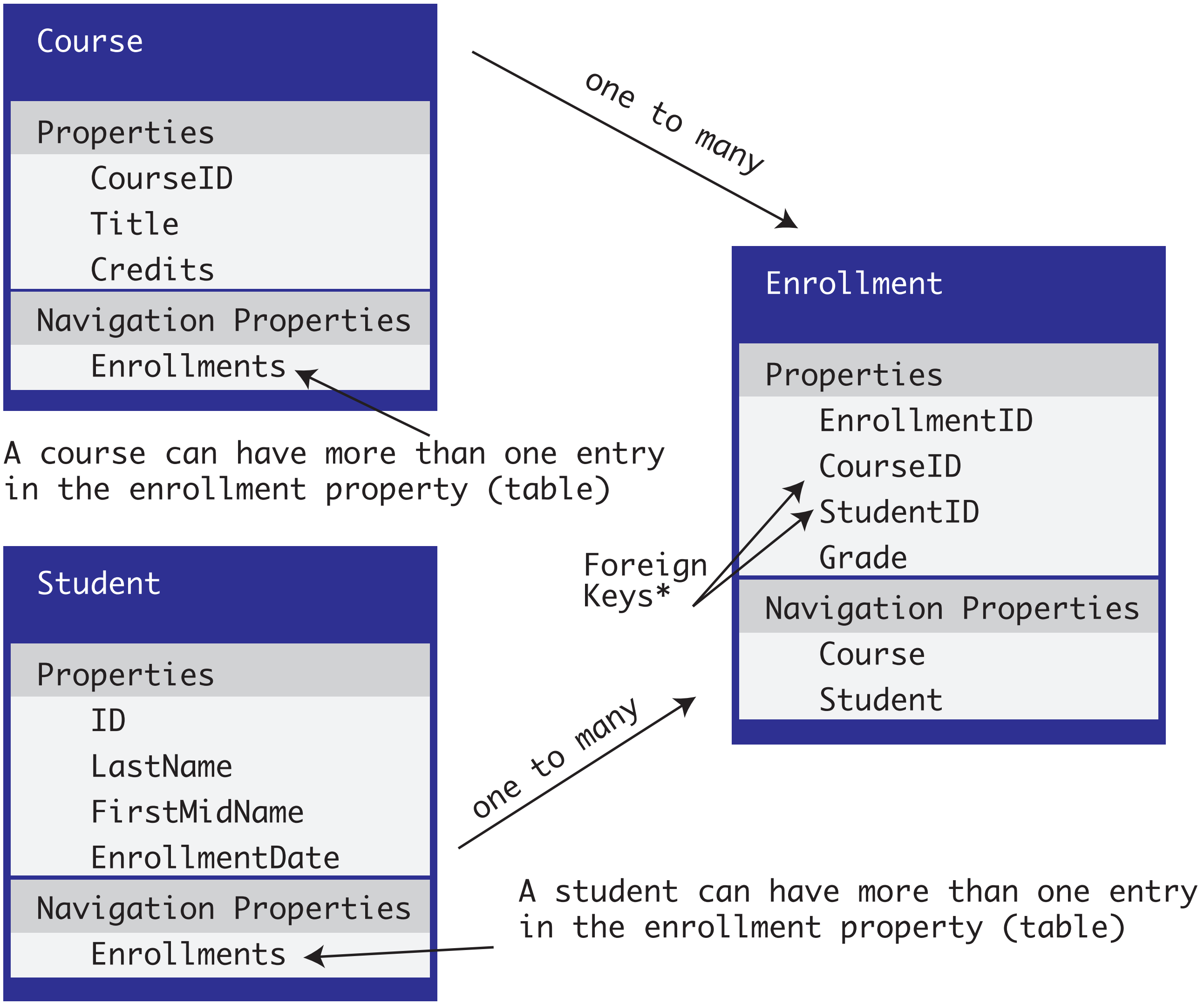
Let’s make a database in MVC



Course

Properties

CourseID

Title

Credits

Navigation Properties

Enrollments

one to many

Enrollment

Properties

EnrollmentID

CourseID

StudentID

Grade

Navigation Properties

Course

Student

A course can have more than one entry
in the enrollment property (table)

Foreign
Keys*

Student

Properties

ID

LastName

FirstMidName

EnrollmentDate

Navigation Properties

Enrollments

one to many

A student can have more than one entry
in the enrollment property (table)

*A note about foreign keys...

EF Core interprets a property as a foreign key if it’s the name of the navigation property (table) followed by the primary key property name (ID). Use StudentID for the Student navigation property, since the Student entity’s primary key is ID, and, similarly, CourseID since the Course entity’s primary key is CourseID.

Now let’s put that in code...

First, make a folder in your project called models. Then, make a seperate file for each property (table). Name it the same as your property and make sure it has a .cs at the end.

Student.cs should look like this:

```
using System;
using System.Collections.Generic;

namespace ContosoUniversity.Models
{
    public class Student
    {
        // ID property can also be StudentID. Either naming
        // convention sets it as the primary key, but pick one.
        public int ID { get; set; }
        public string LastName { get; set; }
        public string FirstMidName { get; set; }
        public DateTime EnrollmentDate { get; set; }

        // Enrollments is a navigation property. Navigation
        // properties link to other entities that are related to
        // this entity. One student can have mulitple Enrollments,
        // so it can hold multiple entities. So the navigation
        // property must be a list type (ie: ICollection)
        public ICollection<Enrollment> Enrollments { get; set; }
    }
}
```

Course.cs should look like this:

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;

namespace ContosoUniversity.Models
{
    public class Course
    {
        // Attribute allows the app to specify the primary key
        // rather than having the DB generate it
        [DatabaseGenerated(DatabaseGeneratedOption.None)]
        public int CourseID { get; set; }
        public string Title { get; set; }
        public string Credits{ get; set; }

        // The Enrollments property is a navigatin property. A Course
        // entity can be related to any number of Enrollemnt entities
        public ICollection<Enrollment> Enrollements { get; set; }
    }
}
```

Here’s how those tables look in code... (cont.)

Enrollment.cs should look like this:

```
namespace ContosoUniversity.Models
{
     public enum Grade
     {
          A, B, C, D, F
     }

     public class Enrollment
     {
          public int EnrollmentID { get; set; }
          // Can have multiple of the same CourseID and studentID in
          // this table. They are foreign keys that specify which
          // course or student.
          public int CourseID { get; set; }
          public int StudentID { get; set; }
          // The ? means this property can be set to null
          public Grade? Grade { get; set; }

          public Course Course { get; set; }
          public Student Student { get; set; }

     }
}
```

Now Put them all in a database together

Make another .cs file in the Models folder to house the database.
Name it SchoolContext.cs that looks like this:

```
using Microsoft.EntityFrameworkCore;

namespace ContosoUniversity.Models
{
    public class SchoolContext : DbContext
    {
        public SchoolContext(DbContextOptions<SchoolContext> options)
            : base(options)
        {
        }

        public DbSet<Student> Student { get; set; }
        public DbSet<Enrollment> Enrollment { get; set; }
        public DbSet<Course> Course { get; set; }
    }
}
```