ForeignKey Attribute in Entity Framework Core

Standard Entity, which serves as our **Principal Entity**, with **StandardId** as the **Primary Key** Property. Note that we have a mandatory collection navigational property called “**students**” that is necessary for implementing **Foreign Key Relationships** using EF Core.

namespace EFCoreCodeFirstDemo.Entities

{

public class Standard

{

public int StandardId { get; set; }

public string? StandardName { get; set; }

public string? Description { get; set; }

public ICollection<Student>? Students { get; set; }

}

}

The **Student Entity**is the Dependent Entity. Note that we have to include the **Standard Reference Navigational Property**, which is mandatory for establishing the Foreign Key in the Student database table. Additionally, we have to include the **StandardId** property, which refers to the **Primary Key** Property of the **Standard** table.

namespace EFCoreCodeFirstDemo.Entities

{

public class Student

{

public int StudentId { get; set; }

public string? FirstName { get; set; }

public string? LastName { get; set; }

//The Following Property Exists in the Standard Entity

public int StandardId { get; set; }

//To Create a Foreign Key it should have the Standard Navigational Property

public Standard? Standard { get; set; }

}

}

In this example, we will demonstrate the **one-to-many** relationship between the Student and Standard Entities. This relationship is represented by including **StandardId** in the **Student class**, which has a reference property called **Standard**. Meanwhile, the **Standard** entity class includes a **collection navigation property called Students**. The **StandardId** property in the Student Class matches the primary key property of the **Standard** class. As a result, the StandardId property in the Student class will automatically become a **Foreign Key Property**, and the corresponding column in the database table will also become a Foreign Key Column.

**Note:**You must remember that both Entities should and must have Navigational Properties to implement Foreign Key Relationships.

Associated Context Class.

using Microsoft.EntityFrameworkCore;

namespace EFCoreCodeFirstDemo.Entities

{

public class EFCoreDbContext : DbContext

{

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer(@"Server=LAPTOP-6P5NK25R\SQLSERVER2022DEV;Database=EFCoreDB;Trusted\_Connection=True;TrustServerCertificate=True;");

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

}

public DbSet<Student>Students { get; set; }

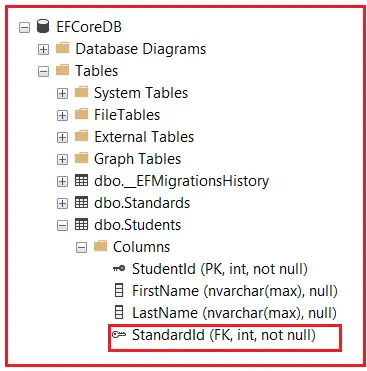
public DbSet<Standard> Standards { get; set; }

}

}

Sync the database with the model using **add-migration** and **update-database** commands using Package Manager Console or .NET Core CLI.

If you verify the database, the Foreign key should have been created in the Student database, as shown in the image below. Here, the Foreign Key Property is created with the name StudentId.



Suppose the **Foreign Key Property** does not exist in the **Dependent Entity** class. In that case, Entity Framework will create a Foreign Key column in the Database table with the **Primary Key Property Name of the Principal Entity**.

Modify the **Student Entity** class as follows. Here, you can see we have not added the **StandardId** property. So, in this case, Entity Framework will create the Foreign Key with the name **StandardId**. We must have added the Standard reference navigation Property. Otherwise, the foreign key will not be created.

namespace EFCoreCodeFirstDemo.Entities

{

public class Student

{

public int StudentId { get; set; }

public string? FirstName { get; set; }

public string? LastName { get; set; }

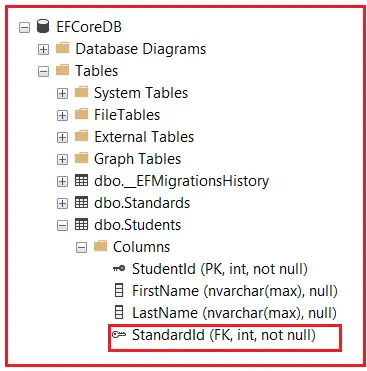
//To Create a Foreign Key it should have the Standard Navigational Property

public Standard? Standard { get; set; }

}

}

With the above changes in place, open Package Manager Console and Execute the following **add-migration** and **update-database** commands.



The **[ForeignKey(name)]** attribute can be applied in three ways:

1. **[ForeignKey(NavigationPropertyName)]** on the Foreign Key Scalar Property in the Dependent Entity.
2. **[ForeignKey(ForeignKeyPropertyName)]** on the Related Reference Navigation Property in the Dependent Entity.
3. **[ForeignKey(ForeignKeyPropertyName)]** on the Collection Navigation Property in the Principal Entity.

##### **[ForeignKey] on the Foreign Key Scaler Property in the Dependent Entity**

The **[ForeignKey]** Attribute in Entity Framework Core can be applied to the Foreign Key Scalar Property in the Dependent Entity. In this case, we must specify the related Navigation Property name in the ForeignKey.

For a better understanding, please modify the Student Entity as follows. As you can see in the below code, we have applied the **[ForeignKey(“Standard”)]** on the StandardReferenceId scaler property, which will be created as the Foreign Key in the database pointing to the Standard table StandardId (Primary Key) column.

using System.ComponentModel.DataAnnotations.Schema;

namespace EFCoreCodeFirstDemo.Entities

{

public class Student

{

public int StudentId { get; set; }

public string? FirstName { get; set; }

public string? LastName { get; set; }

[ForeignKey("Standard")]

public int StandardReferenceId { get; set; }

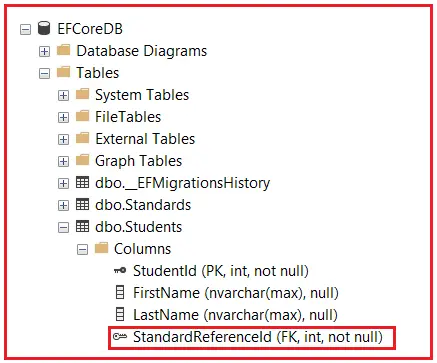
//Related Standard Navigational Property

public Standard? Standard { get; set; }

}

}

With the above changes in place, open the Package Manager Console and Execute **add-migration** and **update-database** commands.



**[ForeignKey] on the Reference Navigation Property in the Dependent Entity**

The **[ForeignKey]** Attribute in EF Core can also be applied to the Reference Navigation Property of the Dependent Entity. In this case, we need to specify the dependent entity foreign key scaler property name in the [ForeignKey] Attribute. For a better understanding, please modify the Student Entity as follows. As you can see, we have StandardReferenceId in the [ForeignKey] Attribute, i.e., **[ForeignKey(“StandardReferenceId”)],** which is applied to the Standard Navigation Property.

using System.ComponentModel.DataAnnotations.Schema;

namespace EFCoreCodeFirstDemo.Entities

{

public class Student

{

public int StudentId { get; set; }

public string? FirstName { get; set; }

public string? LastName { get; set; }

//Related Foreign Key Property

public int StandardReferenceId { get; set; }

[ForeignKey("StandardReferenceId")]

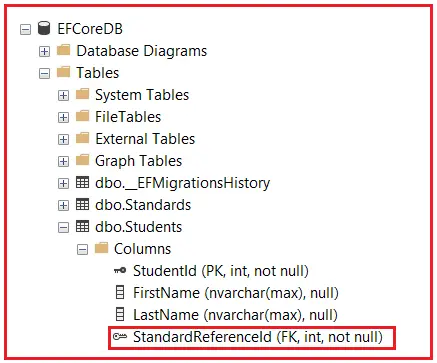
public Standard? Standard { get; set; }

}

}

In the above example, the [ForeignKey] attribute is applied on the Standard navigation property, and the name of the foreign key scaler property StandardReferenceId is specified. Here, Entity Framework Core will create the foreign key column StandardReferenceId in the Students table in the database.

With the above changes, open Package Manager Console and Execute **add-migration** and **update-database** commands.



**[ForeignKey] on the Collection Navigation Property in the Principal Entity**

The [ForeignKey] Attribute in EF Core can also be applied to the Collection Navigation Property in the Principal Entity, and the related foreign key scaler property name can be specified in the Dependent Entity. Let us understand this with an example. First, modify the Student Entity as follows. Here, you can see we have removed the ForeignKey Attribute.

namespace EFCoreCodeFirstDemo.Entities

{

public class Student

{

public int StudentId { get; set; }

public string? FirstName { get; set; }

public string? LastName { get; set; }

public int StandardReferenceId { get; set; }

public Standard? Standard { get; set; }

}

}

We want the Property **StandardReferenceId** to be created as a Foreign Key column in the Students database table, which must be pointed to the **StandardId** Property of the **Standard** entity. To do so, modify the **Standard** Entity as follows. Here, you can see the **[ForeignKey]** attribute is applied on the **Students** collection navigation property, and here, we are passing the **StandardReferenceId** property to the [ForeignKey] attribute, which will make the **StandardReferenceId** a Foreign Key Property.

using System.ComponentModel.DataAnnotations.Schema;

namespace EFCoreCodeFirstDemo.Entities

{

public class Standard

{

public int StandardId { get; set; }

public string? StandardName { get; set; }

public string? Description { get; set; }

[ForeignKey("StandardReferenceId")]

public ICollection<Student>? Students { get; set; }

}

}

With the above changes, open Package Manager Console and Execute **add-migration** and **update-database** commands.

