

Relaciones con Entity Framework

En SQL, las relaciones entre tablas se llaman "relaciones de clave externa" o "foreign key relationships" en inglés.

Una relación de clave externa se crea cuando una tabla en una base de datos tiene una columna que se relaciona con la columna de otra tabla en la misma base de datos. La columna que establece la relación en la tabla es conocida como la "clave externa" o "foreign key", y la columna correspondiente en la otra tabla es conocida como la "clave primaria" o "primary key".

La relación de clave externa se utiliza para asegurarse de que la información en ambas tablas esté sincronizada y para permitir que se realicen consultas complejas que involucren datos de múltiples tablas. Al definir una relación de clave externa, se establece una restricción de integridad referencial que impide la creación de registros huérfanos en la tabla secundaria, es decir, registros que no tienen un registro correspondiente en la tabla principal. Esto ayuda a garantizar la integridad de los datos y a evitar inconsistencias en la base de datos.

Las relaciones más usadas en base de datos relaciones son:

Uno a uno (One-to-One): Una entidad está asociada con exactamente una instancia de otra entidad. En este caso, cada entidad tiene una clave primaria que también es una clave foránea hacia la otra entidad.

Uno a muchos (One-to-Many): Una entidad está asociada con varias instancias de otra entidad. En este caso, la entidad que tiene varias instancias tiene una colección de la otra entidad.

Muchos a muchos (Many-to-Many): Varias instancias de una entidad están asociadas con varias instancias de otra entidad. En este caso, se requiere una tabla intermedia para almacenar las relaciones entre las dos entidades.

Y con Entity Framework podemos crearlas fácilmente. Tenemos el siguiente caso de ejemplo que podría aplicar los 3 tipos de relaciones vistos:

Vehículos

Motores

Marcas

Accesorios

El vehículo tiene solo un motor y el motor solo tiene un vehículo (número de motor único) (**One-to-One**).

El vehículo tiene una marca y una marca puede tener muchos vehículos

(One-to-Many)

El vehículo puede tener muchos accesorios y los accesorios lo pueden tener muchos vehículos **(Many-to-Many)**

Ejemplos de los tres tipos de relaciones

```
public class Car
{
    public int Id { get; set; }
    public int MotorId { get; set; }
    public Motor Motor { get; set; }
    public List<Accessory> Accessories { get; } = new List<Accessory>();
}

public class Motor
{
    public int Id { get; set; }
    public int Number
    public string Descripcion
    public int CarId { get; set; }
    public virtual Car Car { get; set; }
    public int BrandId { get; set; }
    public virtual Brand Brand { get; set; }
}

public class Brand
{
    public int Id { get; set; }
    public int Name { get; set; }
    public virtual ICollection<Car> Cars { get; } = new List<Car>();
}

public class Accessory
{
    public int Id { get; set; }
    public int Name { get; set; }
    public decimal Price { get; set; }
    public List<Car> Cars { get; } = new List<Car>();
}
```

Adicionalmente leer la documentación de Microsoft sobre las relaciones en EF. Comprobarán que no hay una única forma de definir las relaciones y a veces depende de las necesidades del proyecto o las convenciones que se están utilizando.

<https://learn.microsoft.com/es-es/ef/core/modeling/relationships>