

M\$ EF - Code First

NuGet package manager

microsoft.EntityFrameworkCore
microsoft.EntityFrameworkCore.Design
microsoft.EntityFrameworkCore.Tools

Models

Skapa mappen entities (Filmen säger models, men Hans indikerar att det är bäst att hålla isär)

Skapa klassen **Product** (med MS Snippet: Ta bort product-klassen och Börja skriv **Product**. Använd *snippet* för att få VS att skriva ut hela klassen.

option: Markera att **Id** är en PK genom [KEY] (men det behövs inte för just nyckelordet **Id**)

Notera att **Name** är nödvändigt med [Required]

Notera också att **Price** är typ satt till Decimal med två decimalers precision.

product:

```
public class Product
{
    public int Id { get; set; }    //Id blir automagiskt PK

    [Required]
    public string Name { get; set; }

    [Column (TypeName = "decimal(18, 2)") ]
    public decimal Price { get; set; }
}
```

Skapa klassen **Customer**

Notera att här används #-direktiv för att slå på och av c# nullable. OM jag fattar rätt, kan man göra undantag genom att sätt frågetecken efter typen, som vid **Address** och **Phone** nedan:

Sista genererade koden rör klassen order, som enklast skapas med intellisense ...
Denna skapar ett "one 2 many" förhållande (Men jag fattar inte hur!)

Customer

```
public class Customer
{
    #nullable enable

    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string? Address { get; set; }
    public string? Phone { get; set; }

    #nullable disable

    public ICollection<Order> Orders { get; set; }
}
```

Resultterande kod för Order

```
public class Order
{
    public int Id { get; set; }
    public DateTime OrderPlaced { get; set; }
    public DateTime? OrderFulfilled { get; set; }
    public int CustomerId { get; set; }
    public Customer Customer { get; set; }
    public ICollection<ProductOrder> Productorders { get; set; }
}
```