Ctrl + ; = comenta TrustServerCertificate=True"

"Server=localhost,1433; Database=balta; User ID=sa; Password=1q2w3e4r@#$; TrustServerCertificate=True";

Alt + Shit + F = Identação  
Alt + Seta pra cima = carrega linha  
Alt + seta pra baixo = carrega pra baixo  
Ctl +Sht + Enter = pula linha de cima  
Alt + SShit + seta pra baixo = duplicado linha  
Ctrl + D = Linha se repete  
Ctl + B = Barra do lado abre e fecha  
  
Pacotes  
dotnet add package Microsoft.EntityFrameWorkCore.SqlServer  
dotnet add package Microsoft.EntityFrameWorkCore

MODELS X ENTITY

As duas representam objetos que vao para o banco de dados, mas o Intity, possuem pontos a mais.

TABELA X COLUNA DA TABELA

Tela de celular com fundo preto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Data Contex é a representação do bando de dados em memória, possuem os DBset e refletem as tabelas.  
  
Criado pasta e arquivo DataContex.CS

Mapeamento das Categorias D = Para

Seu local

1º criado categoria em model  
Texto

Descrição gerada automaticamente

2º criado pasta Data – data contex representa o bando de memoria e Categories representa tabela de categorias  
Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

3º Em programa já é possível instanciar a categoria e fazer o CRUDE  
Texto

Descrição gerada automaticamente

* **Configurando o DataContext**

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

* **Create**

using Blog.Models;

using Microsoft.EntityFrameworkCore;

namespace Blog.Data

{

    public class BlogDataContext : DbContext

    {

        public DbSet<Category> Categories { get; set; }

        public DbSet<Post> Posts { get; set; }

        // public DbSet<PostTag> PostTags { get; set; }

        public DbSet<Role> Roles { get; set; }

        public DbSet<Tag> Tags { get; set; }

        public DbSet<User> Users { get; set; }

        // public DbSet<UserRole> UserRoles { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder options)

            => options.UseSqlServer("Server=localhost,1433; Database=Blog; User ID=sa; Password=1q2w3e4r@#$; TrustServerCertificate=True");

    }

}

namespace Blog.Models

{

    public class Category //TABELA

    {

    public int? Id { get; set; }  // COLUNAS

    public string? Name { get; set; }

    public string? Slug { get; set; }

    }

}

namespace Blog.Models

{

    public class Post //TABELA

    {

    public int Id { get; set; }

    public int CategoryId { get; set; }

    public int AuthorId { get; set; }

    public string? Title { get; set; }

    public string? Summary { get; set; }

    public string? Body { get; set; }

    public string? Slug { get; set; }

    public DateTime CreateDate { get; set; }

    public DateTime LastUpdateDate { get; set; }

    }

}

namespace Blog.Models

{

    public class PostTag //TABELA

    {

    public int? PostId { get; set; }  // COLUNAS

    public string? TagId { get; set; }

    }

}

namespace Blog.Models

{

    public class Role //TABELA

    {

    public int? Id { get; set; }  // COLUNAS

    public string? Name { get; set; }

    public string? Slug { get; set; }

    }

}

//   CONSTRAINT [PK\_Role] PRIMARY KEY([Id]),

//     CONSTRAINT [UQ\_Role\_Slug] UNIQUE([Slug])

// )

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace Blog.Models

{

    [Table ("Tag")]

    public class Tag //TABELA

    {

    [Key]

    public int? Id { get; set; }  // COLUNAS

    public string? Name { get; set; }

    public string? Slug { get; set; }

    }

}

namespace Blog.Models

{

    public class User   // TABELA

    {

    public int Id { get; set; }  // COLUNA DA TABELA

    public string? Name { get; set; }

    public string? Emai { get; set; }

    public string? PasswordHash { get; set; }

    public string? Image { get; set; }

    public string? Slug { get; set; }

     public string? Bio { get; set; }

    }

}

namespace Blog.Models

{

    public class UserRole //TABELA

    {

    public int? UserId { get; set; }  // COLUNAS

    public string? RoleId { get; set; }

    }

}

�using System;

using System.Linq;

using Blog.Data;

using Blog.Models;

using Microsoft.EntityFrameworkCore;

namespace Blog

{

    class Program

    {

        static void Main(string[] args)

        {

            using (var context = new BlogDataContext())

            {

                 var tag = new Tag { Name = ".NETdvdsvtttt", Slug = "dotnet" };

                 context.Tags.Add(tag);

                 context.SaveChanges();

            }

        }

    }

}

Tela de computador com ícones coloridos

Descrição gerada automaticamente