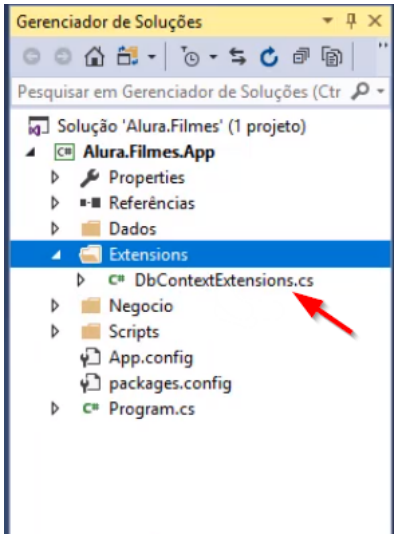
07

Configurando nomes de tabelas

Transcrição

Iremos quebrar a convenção do Entity que utiliza o nome da propriedade `DbSet` como nome de tabela, pois essa regra não é interessante para o projeto do nosso banco de dados legado. Para analisarmos de que forma Entity está gerando o SQL e como está tentando enviá-lo para o banco de dados, utilizaremos um método denominado `LogSQLToConsole` . Trata-se de um método de extensão, disponível na classe `DbContestExtensions.cs` . Basicamente, esse código é o mesmo que fizemos no **primeiro curso** para logar o SQL do Entity. Encapsulamos esse código em um método de extensão e podemos utiliza-lo neste momento.



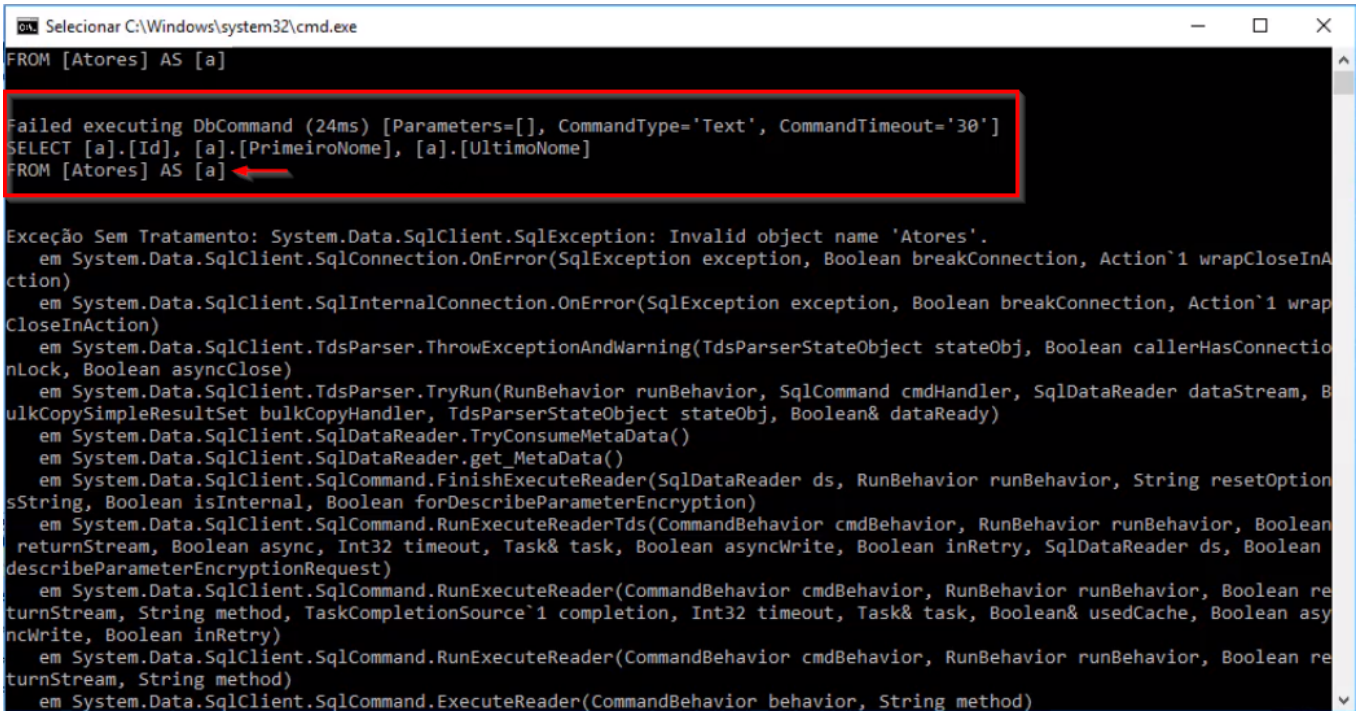

```
using Alura.Filmes.App.Dados;
using Alura.Fiomes.App.Extensions;

namespace Alura.Filmes.App
{
    class Program
    {
        static void Main(string[] args)
        {
            //select * from actor
            using (var contexto =  new AluraFilmesContexto())
            {
                contexto.LogSQLToConsole();

                foreach (var ator in contexto.Atores)
                {
                    System.Console.WriteLine(ator);
                }
            }
        }
    }
}
```

Tentaremos iniciar a aplicação e novamente teremos o erro. Também será exibido o SQL que o Entity está tentando enviar para o banco de dados. O programa falhou ao executar o comando selecionado, e o problema está na "FROM [Atores] As [a]".

Teremos, portanto, de quebrar a convenção do Entity de utilizar a propriedade `DbSet` como nome de tabela.

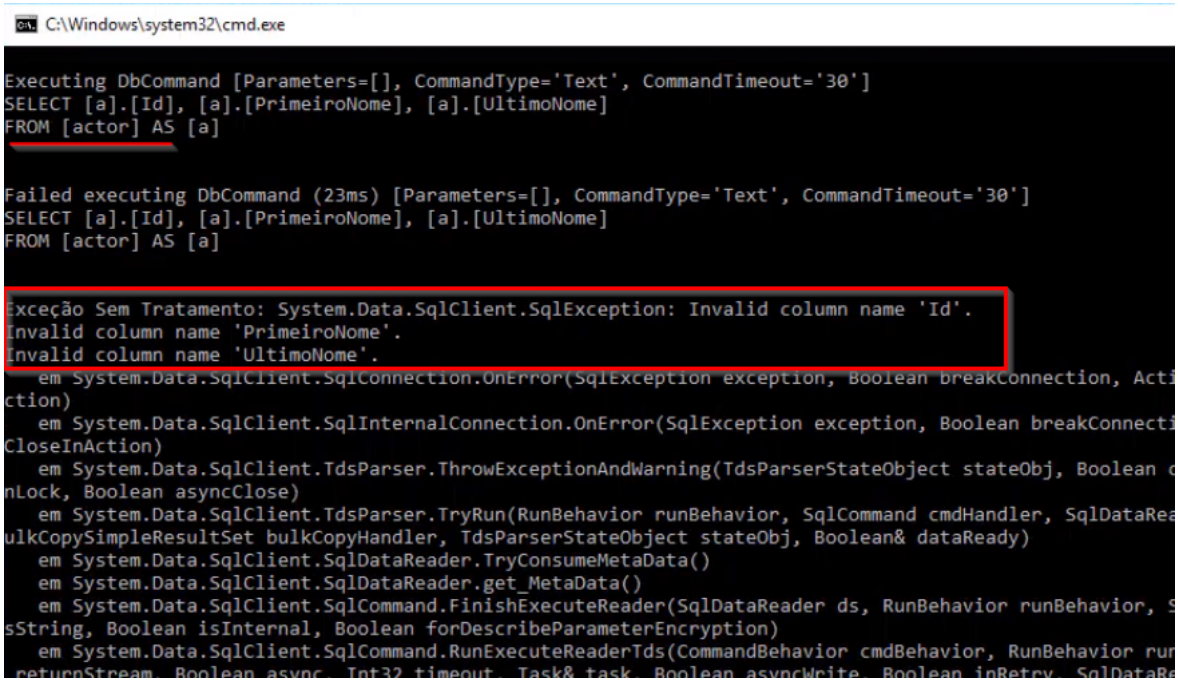


Existem duas maneiras ara quebrarmos a convenção do Entity. A primeira delas envole decorarmos a classe `Ator` com uma anotação denominada `Table`. Nessa anotação, passaremos um argumento para o construtor da classe com o nome da tabela que deve ser mapeada. O nome da tabela, no caso, é `actor`.

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

namespace Alura.Filmes.App.Negocio
{
    [Table("actor")]
    public class Ator
    {
        public int Id { get; set; }
        public string PrimeiroNome { get; set }
        public string UltimoNome { get; set }
    }
}
```

Tentaremos executar a aplicação e receberemos novamente a mensagem de erro. Observaremos que o erro é de uma diferente natureza, dessa vez o nome da tabela (`actor`) está correto, como podemos ver. O erro está relacionado às colunas `Id` , `PrimeiroNome` e `UltimoNome` .



Ao analisarmos o `select` que foi montado, veremos que está usando o nome das colunas. Essa é a segunda convenção do Entity que aprendemos: o *sofware* utiliza o nome da propriedade que está na classe para mapear o nome da coluna na tabela. Teremos de quebrar essa regra.