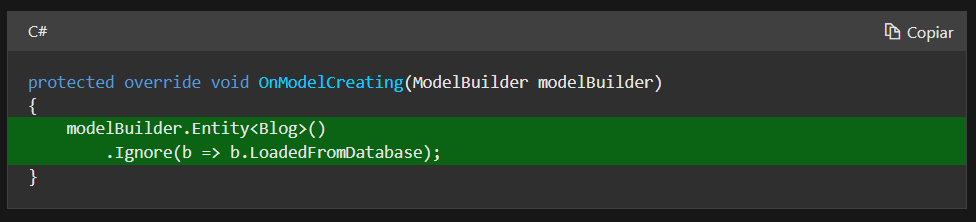
Entity Framework Core parte 2: Mapeando um banco pré-existente

* Concluindo o mapeamento da tabela de atores:
  + Para isso, há a seguinte regra: para toda chave estrangeira, ele criará um índice.
  + O **EF Core** cria um índice para cada chave estrangeira encontrada.
  + O nome estabelecido para os índices encontrados no modelo de dados é definido usando a convenção **idx\_<classe que contém o índice>\_<nome da coluna>**.
  + Não é possível configurar índices usando anotações.
  + A restrição **Unique** é chamada de **Alternate Keys**. Como se fosse a chame primária, só que alternativa.
  + Quando o comando Add-Migration com a configuração da restrição foi executado, o Entity a incluiu em seu modelo de dados (via snapshot). No momento em que o método AddRange() foi chamado, o modelo foi validado, e uma de suas regras (no caso a restrição unique) não foi cumprida, levando ao lançamento da exceção.
* Classificação indicativa de filmes:
  + **Entity** não dá suporte nativo a restrições do tipo **CHECK** e aprendeu a criá-las usando o recurso de **Migrations** do **EF Core**, através do método **Sql()**.
  + Por convenção, todas as propriedades públicas com um **getter** e um **setter** serão incluídas no modelo. Propriedades específicas podem ser excluídas da seguinte maneira: (<https://docs.microsoft.com/pt-br/ef/core/modeling/entity-properties?tabs=fluent-api%2Cwithout-nrt>)



* Mapeando herança:
  + As convenções do **Entity** para mapeamento de herança, são as seguintes:
    - **TPH**, ou ***Table Per Hierarchy***:
      * No qual é criada apenas uma tabela por hierarquia. Será selecionada a classe mais ancestral, e criada uma tabela que incorpore todas as colunas de seus descendentes. Além disso, será criada mais uma coluna, para discriminar a origem de determinado registro.
    - **TPC**, ou ***Table Per Concrete Type***:
      * Cria uma tabela para cada classe concreta. Este método identifica as classes filhas e cria uma tabela para cada uma delas.
    - **TPT**, ou ***Table Per Type***:
      * É criada uma tabela para todos os tipos, ou seja, todos aqueles que fazem parte da hierarquia terão uma.
  + O **EF** Core ainda **NÃO** suporta o segundo e terceiro padrões, ou seja, o **TPC** e **TPT**. Dentro de uma hierarquia, ele trabalhará somente com o **TPH**, criando uma única tabela.
* Assumindo o controle da geração do **SQL**:
  + **FromSql** e suas limitações:
    - <https://docs.microsoft.com/pt-br/ef/core/querying/raw-sql>
  + A query **SQL** deve retornar valores para todas as propriedades da entidade.
  + Uma query executada pelo **FromSql()** não pode conter dados relacionados.
  + Queries **SQL** só podem ser utilizadas para retornar tipos que são entidades, ou seja, tipos que estão sendo gerenciados pelo **EF**.
  + As colunas que são listadas no **SELECT** devem possuir o mesmo nome que o nome das propriedades da entidade.
* Stored Procedures:
  + ExecuteSqlCommand:
    - Executa o comando **DDL/DML** fornecido no banco de dados.