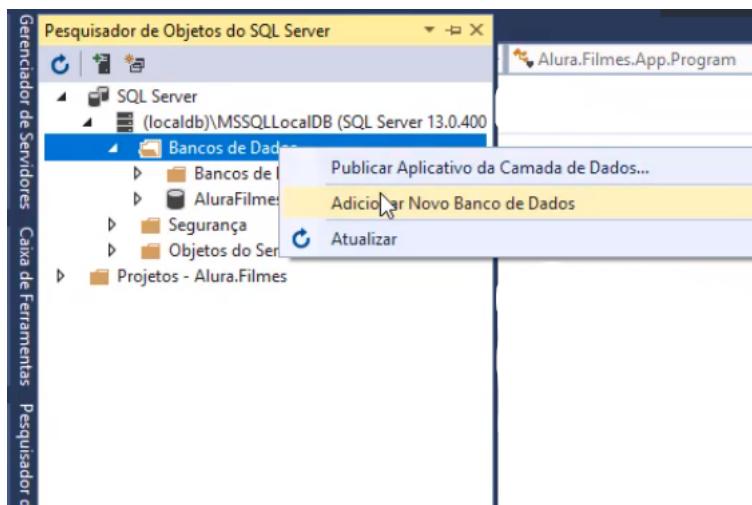


01

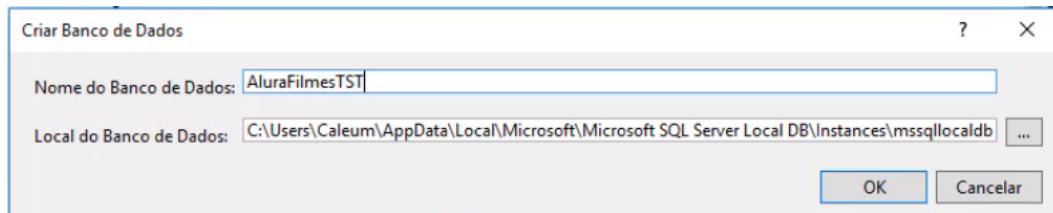
Gerando o script do banco de dados

Transcrição

Conseguimos listar os atores, mas precisamos analisar a estrutura da tabela. Para isso, replicaremos a estrutura de tabela em outro banco de dados por via do próprio servidor local. Na área "Pesquisador de Objetos", selecionaremos o "SQL Server > (localdb) > Banco de dados", clicaremos com o botão direito em "Banco de dados" e acionaremos "Adicionar Novo Banco de Dados".



Chamaremos esse novo banco de dados de `AluraFilmesTST`.



O novo banco de dados está vazio, portanto, utilizaremos o Entity para gerar a estrutura da tabela `Autor`. Assim, poderemos verificar se a estrutura da tabela reflete o banco de dados atual. Usaremos um recurso chamando `migration`. Tal recurso nos ajuda a atualizar a estrutura do banco de dados a partir das mudanças que estão sendo realizadas nas classes. O `migration` vem em um pacote separado dos tradicionais do Entity Framework Core, mas não precisaremos instalá-lo, pois ele já está inserido no pacote do projeto inicial. Para vocês terem certeza de que o recurso está instalado basta digitar o comando `get-help entityframework` na área do console. Caso o recurso esteja instalado corretamente, surgirá um texto de confirmação.

The screenshot shows the 'Console do Gerenciador de Pacotes' window. The command `get-help entityframework` was run, and the output shows the Entity Framework logo followed by the text 'TOPIC about_EntityFrameworkCore'.

```
PM> get-help entityframework
      / \
     / \ 
    / \ 
   / \ 
  / \ 
 / \ 
TOPIC about_EntityFrameworkCore
```

A tabela de atores está declarada da seguinte forma:

```
CREATE TABLE actor (
    actor_id int NOT NULL IDENTITY ,
    first_name VARCHAR(45) NOT NULL,
    last_name VARCHAR(45) NOT NULL,
    last_update DATETIME NOT NULL,
    CONSTRAINT pk_actor PRIMARY KEY NONCLUSTERED (actor_id)
)
```

A tabela original difere do script gerado pelo Entity.

```
CREATE TABLE [actor] (
    [actor_id] int NOT NULL IDENTITY
    [first_name] nvarchar(max) NULL,
    [last_name] nvarchar(max) NULL,
    CONSTRAINT [PK_actor] PRIMARY KEY ([actor_id])
);
```

Mais especificamente, os códigos se diferenciam no que diz respeito a determinação do tipo e tamanho das colunas (`nvarchar(max)`). O Entity utilizou uma de suas convenções, e teremos de reconfigurar de forma mais autônoma o tipo de tamanho das colunas no banco de dados.