

Configurando o mapeamento de filmes

Transcrição

O erro `Invalid object name 'Filmes'` é resultado da convenção utilizada pelo Entity para definir nome de tabela, lembrando que essa convenção consiste em utilizar o nome da propriedade `DbSet`, que neste caso, é `Filmes`. Como essa convenção não atende aos nossos interesses, iremos configurar o mapeamento da tabela `Filmes`.

Faremos as modificações acionando a boa prática de deixar a classe de negócio bem organizada, ou seja, não iremos misturar o mundo relacional com o mundo do objeto. Configuraremos a classe `Filme` no método `OnModelCreating()`. Mapearemos a entidade `Filme` para a tabela `film`.

```
namespace Alura.Filmes.App.Dados
{
    public class AluraFilmesContexto : DbContext
    {
        public DbSet<Ator> Atores { get; set; }
        public DbSet<Filme> Filmes { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer("Server=(localdb)mssqllocaldb;Database=AluraFilmes;Trusted_Connection=true");
        }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<Ator>()
                .ToTable("actor");

            modelBuilder.Entity<Ator>()
                .Property(a => a.Id)
                .HasColumnName("actor_id");

            modelBuilder.Entity<Ator>()
                .Property(a => a.PrimieroNome)
                .HasColumnName("first_name")
                .HasColumnType("varchar(45)")
                .IsRequired();

            modelBuilder.Entity<Ator>()
                .Property(a => a.UltimoNome)
                .HasColumnType("varchar(45)")
                .IsRequired();

            modelBuilder.Entity<Ator>()
                .Property<DateTime>("last_update");
                .HasColumnType("datetime")
                .HasDefaultValueSql("getdate()")
                .IsRequired();

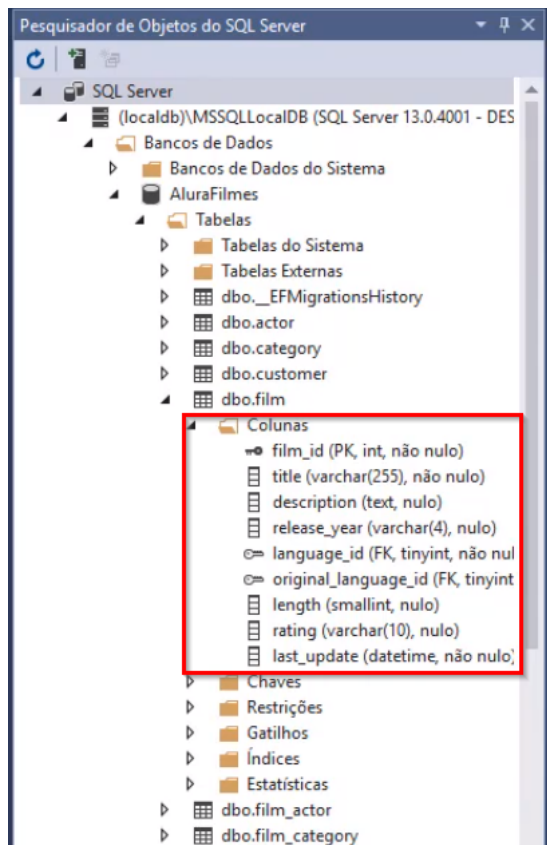
            modelBuilder.Entity<Filme>()
                .ToTable("film")
```

```

    }
}
}

```

Feito isso, poderemos começar a mapear as propriedades nas colunas (Id , Titulo , Descricao , AnoLancamento , Duracao). Também iremos configurar os tipos das colunas e sua nulidade - esses dados são obtidos de acordo com informações exibidas ao lado direito da tela, na área "Pesquisador de Objetos". -



```

namespace Alura.Filmes.App.Dados
{
    public class AluraFilmesContexto : DbContext
    {
        public DbSet<Ator> Atores { get; set; }
        public DbSet<Filme> Filmes { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer("Server=(localdb)mssqllocaldb;Database=AluraFilmes;Trusted_Connection=true;");
        }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<Ator>()
                .ToTable("actor");

            modelBuilder.Entity<Ator>()
                .Property(a => a.Id)
                .HasColumnName("actor_id");

            modelBuilder.Entity<Ator>()

```

```

        .Property(a => a.PrimieroNome)
        .HasColumnName("first_name")
        .HasColumnType("varchar(45)")
        .IsRequired();

modelBuilder.Entity<Ator>()
    .Property(a => a.UltimoNome)
    .HasColumnType("varchar(45)")
    .IsRequired();

modelBuilder.Entity<Ator>()
    .Property<DateTime>("last_update");
    .HasColumnType("datetime")
    .HasDefaultValueSql("getdate()")
    .IsRequired();

modelBuilder.Entity<Filme>()
    .ToTable("film")

modelBuilder.Entity<Filme>()
    .Property(f => f.id)
    .HasColumnName("film_id");

modelBuilder.Entity<Filme>();
    .Property(f => f.Titulo)
    .HasColumnName("title")
    .HasColumnType("varchar(255)")
    .IsRequired();

modelBuilder.Entity<Filme>();
    .Property(f => f.Descricao)
    .HasColumnName("description")
    .HasColumnType("text");

modelBuilder.Entity<Filme>();
    .Property(f => f.AnoLancamento)
    .HasColumnName("release_year")
    .HasColumnType("varchar(4)");

modelBuilder.Entity<Filme>()
    .Property(f => f.Duracao)
    .HasColumnName("length")

modelBuilder.Entity<Filme>();
    .Property<DateTime>("last_update")
    .HasColumnType("datetime")
    .IsRequired();

    }
}
}

```

Ao executarmos o programa, perceberemos que haverá uma mensagem de erro. O programa fez o mapeamento correto de todas as propriedades, mas vemos a mensagem de erro An exception occurred while reading a database value for

property 'Film.Duracao'. The expected type was 'System.Int32' but the actual value was of type 'System.Int16'. O que podemos fazer para resolver este erro?