

03

3 - Linq to entities refinando consultas

Transcrição

Nosso cliente fez um novo pedido: criar uma consulta que deve trazer as faixas de músicas a partir de dois parâmetros. O primeiro deles é o nome do artista e o segundo o título do álbum.

Vamos iniciar com um contexto:

```
using (var contexto = new AluraTunesEntities) {  
}
```

Em seguida, vamos inserir uma consulta simples:

```
var query = from f in contexto.Faixas
```

Além do `select f` e o `foreach(var faixa in query)`. Ainda, vamos adicionar `Console.WriteLine()` e passamos para ele a formatação, "`{0}\t{1}`", `faixa.Album.Titulo`, `faixa.Nome` e abaixo disso acrescentamos o `Console.ReadKey()`. Teremos:

```
class Program  
{  
    static void Main(string[] args)  
    {  
        using (var contexto = new Alura Tunes Entities())  
        {  
            var query = from f in contexto.Faixas  
            select f;  
  
            foreach(var faixa in query)  
            {  
                Console.WriteLine("{0}\t{1}", faixa.Album.Titulo, faixa.Nome);  
  
            }  
  
            Console.ReadKey();  
        }  
    }  
}
```

Rodando o código, teremos como resultado todas as faixas disponíveis no sistema!

Portanto, o próximo passo é filtrar por nome de artista! Em seguida, adicionaremos o `where` e como não existe na entidade faixas o "Nome do artista", navegaremos até encontrar o `Artista`:

```
where f.Album.Artista.Nome.Contains()
```

Passaremos uma variável que será utilizada na busca: a `buscaArtista`. Como a variável ainda não foi declarada, faremos isso escrevendo:

```
var buscaArtista = "Led Zeppelin"
```

O trecho do código ficará assim:

```
class Program
{
    static void Main(string[] args)
    {
        using (var contexto = new Alura Tunes Entities())
        {
            var buscaArtista = "Led Zeppelin";

            var query = from f in contexto.Faixas
                        where f.Album.Artista.Nome.Contains(buscaArtista)
                        select f;

            foreach(var faixa in query)
            {
                Console.WriteLine("{0}\t{1}", faixa.Album.Titulo, faixa.Nome);
            }
        }

        Console.ReadKey();
    }
}
```

Ao executarmos o código, temos o seguinte resultado:

```
file:///C:/Users/caelum/Documents/Visual Studio 2015/Projects/AluraTunes/bin/Debug/AluraTunes.EXE
Led Zeppelin III      Tangerine
Led Zeppelin III      That's The Way
Led Zeppelin III      Bron-Y-Aur Stomp
Led Zeppelin III      Hats Off To (Roy) Harper
Physical Graffiti [Disc 2]  In The Light
Physical Graffiti [Disc 2]  Bron-Yr-Aur
Physical Graffiti [Disc 2]  Down By The Seaside
Physical Graffiti [Disc 2]  Ten Years Gone
Physical Graffiti [Disc 2]  Night Flight
Physical Graffiti [Disc 2]  The Wanton Song
Physical Graffiti [Disc 2]  Boogie With Stu
Physical Graffiti [Disc 2]  Black Country Woman
Physical Graffiti [Disc 2]  Sick Again
Presence      Achilles Last Stand
Presence      For Your Life
Presence      Royal Orleans
Presence      Nobody's Fault But Mine
Presence      Candy Store Rock
Presence      Hots On For Nowhere
Presence      Tea For One
The Song Remains The Same (Disc 1)  Rock & Roll
The Song Remains The Same (Disc 1)  Celebration Day
The Song Remains The Same (Disc 1)  The Song Remains The Same
The Song Remains The Same (Disc 1)  Rain Song
The Song Remains The Same (Disc 1)  Dazed And Confused
The Song Remains The Same (Disc 2)  No Quarter
The Song Remains The Same (Disc 2)  Stairway To Heaven
The Song Remains The Same (Disc 2)  Moby Dick
The Song Remains The Same (Disc 2)  Whole Lotta Love
```

A tela mostra todas as músicas do "Led Zeppelin" e também os álbuns. Porém, repare que a distribuição dos resultados está estranha, sendo exibidos apenas em duas colunas. Para arrumar a formatação dos elementos, vamos modificar a saída, dest

maneira, adicionaremos no `Console.WriteLine()` a faixa.Album.Titulo e o `PadRight()` - para este último, passaremos o valor `40`, isto significa que o resultado deve ser fixado em `40` colunas:

```
Console.WriteLine("{0}\t{1}", faixa.Album.Titulo.PadRight(40), faixa.Nome);
```

Observe a nova arrumação:

```

file:///C:/Users/caelum/Documents/Visual Studio 2015/Projects/AluraTunes/bin/Debug/AluraTunes.EXE

BBC Sessions [Disc 1] [Live] You Shook Me
BBC Sessions [Disc 1] [Live] I Can't Quit You Baby
BBC Sessions [Disc 1] [Live] Communication Breakdown
BBC Session [Disc 1] [Live] Dazed and Confused
BBC Sessions [Disc 1] [Live] The Girl I Love She Got Long Black Wavy Hair
BBC Sessions [Disc 1] [Live] What is and Should Never Be
BBC Sessions [Disc 1] [Live] Communication Breakdown(2)
BBC Sessions [Disc 1] [Live] Travelling Riverside Blues
BBC Sessions [Disc 1] [Live] Whole Lotta Love
BBC Sessions [Disc 1] [Live] Somethin' Else
BBC Sessions [Disc 1] [Live] Communication Breakdown(3)
BBC Sessions [Disc 1] [Live] I Can't Quit You Baby(2)
BBC Sessions [Disc 1] [Live] You Shook Me(2)
BBC Sessions [Disc 1] [Live] How Many More Times
Physical Graffiti [Disc 1] Custard Pie
Physical Graffiti [Disc 1] The Rover
Physical Graffiti [Disc 1] In My Time Of Dying
Physical Graffiti [Disc 1] Houses Of The Holy
Physical Graffiti [Disc 1] Trampled Under Foot
Physical Graffiti [Disc 1] Kashmir
BBC Sessions [Disc 2] [Live] Immigrant Song
BBC Sessions [Disc 2] [Live] Heartbreaker
BBC Sessions [Disc 2] [Live] Since I've Been Loving You
BBC Sessions [Disc 2] [Live] Black Dog
BBC Sessions [Disc 2] [Live] Dazed And Confused
BBC Sessions [Disc 2] [Live] Stairway To Heaven
BBC Sessions [Disc 2] [Live] Going To California
BBC Sessions [Disc 2] [Live] That's The Way
BBC Sessions [Disc 2] [Live] Whole Lotta Love (Medley)
BBC Sessions [Disc 2] [Live] Thank You

```

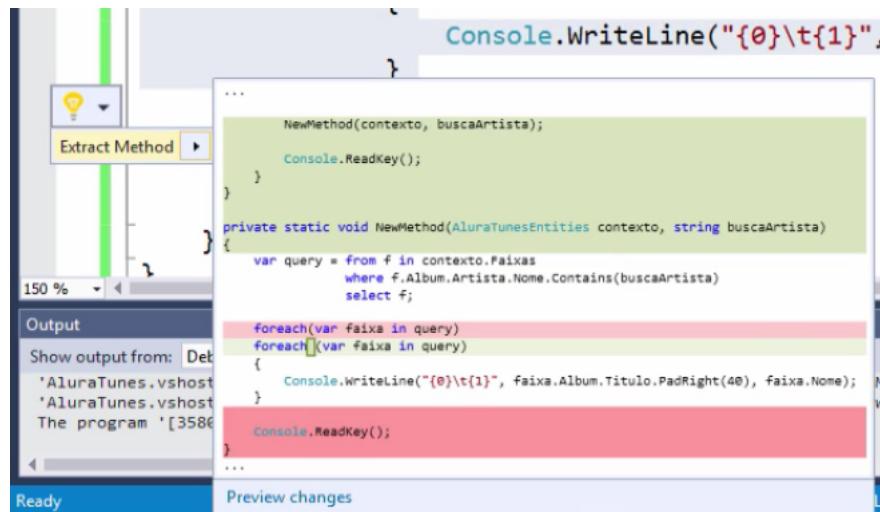
Após resolver o primeiro requisito do cliente, podemos passar a segunda parte do pedido. Ele também deseja um filtro por título de `Album`. Para isso, vamos extrair uma função, assim, selecionaremos o seguinte trecho do código:

```
var query = from f in contexto.Faixas
            select f;

foreach(var faixa in query)
{
    Console.WriteLine("{0}\t{1}", faixa.Album.Titulo, faixa.Nome);

}
```

E vamos extrair a função utilizando "Ctrl ." e depois selecionamos a opção "Extrair método":



Desta maneira, um método é criado com o nome original de `NewMethod()` e iremos renomeá-lo para `GetFaixas()`. Passaremos para ele o parâmetro `buscaAlbum` e acima dele declaramos uma string vazia, a `var buscaAlbum = ""`:

```
using (var contexto = new AluraTunesEntities())
{
    var buscaArtista = "Led Zeppelin";
    var buscaAlbum = "";

    GetFaixas(contexto, buscaArtista, buscaAlbum);

    Console.ReadKey();
}
```

É preciso também passar o `private static void` e ao final da declaração de assinatura do método `GetFaixas()`, teremos `string buscaAlbum`:

```
private static void GetFaixas(AluraTunesEntities contexto, string buscaArtista, string buscaAlbum)
```

Não haverá nenhuma diferença no resultado! Isso ocorre pois ainda não modificamos a consulta.

Primeiro, vamos verificar se o parâmetro `buscaAlbum` está preenchido ou se é uma string vazia. Para descobrir isso acrescentaremos uma condição, de "se `buscaAlbum` for vazio ou nulo nada será filtrado!". Desta forma, incluiremos a seguinte linha:

```
if (!string.IsNullOrEmpty(buscaAlbum))
```

Caso ele não esteja vazio, o programa deve executar a instrução de modificar a `query` deve ser modificada. Para isto, adicionaremos

```
query = query.Where()
```

Passaremos a expressão `lambda`, que receberá um argumento de busca. Por isso, passaremos o parâmetro `q`:

```
q => q.Album.Titulo.Contains(buscaAlbum)
```

O resultado será o seguinte:

```
if (!string.IsNullOrEmpty(buscaAlbum))
{
    query = query.Where(q => q.Album.Titulo.Contains(buscaAlbum));
}
```

A query será modificada apenas se o buscaAlbum possuir algum valor. Mas, rodando a aplicação, teremos um resultado igual ao anterior, pois nada foi filtrado. Assim, apagaremos o buscaArtista e o buscaAlbum e vamos colocar "Led Zeppelin" dentro do GetFaixas(). Teremos:

```
using (var contexto = new AluraTunesEntities())
{
    GetFaixas(contexto, "Led Zeppelin", "");

    Console.ReadKey();
}
```

Ainda assim o resultado não trará nada de diferente! Por isso, vamos copiar novamente o GetFaixas(contexto, "Led Zeppelin", ""); e passamos em vez do vazio, o Graffiti . Teremos:

```
GetFaixas(contexto, "Led Zeppelin", "Graffiti")
```

E entre os GetFaixa() s, vamos utilizar o Console.WriteLine():

```
GetFaixas(contexto, "Led Zeppelin", "");

Console.WriteLine();

GetFaixas(contexto, "Led Zeppelin", "Graffiti");

Console.ReadKey();
```

Ao executarmos, temos o seguinte:

file:///C:/Users/caelum/Documents/Visual Studio 2015/Projects/AluraTunes/bin/Debug/AluraTunes.EXE	
Presence	Nobody's Fault But Mine
Presence	Candy Store Rock
Presence	Hots On For Nowhere
Presence	Tea For One
The Song Remains The Same (Disc 1)	Rock & Roll
The Song Remains The Same (Disc 1)	Celebration Day
The Song Remains The Same (Disc 1)	The Song Remains The Same
The Song Remains The Same (Disc 1)	Rain Song
The Song Remains The Same (Disc 1)	Dazed And Confused
The Song Remains The Same (Disc 2)	No Quarter
The Song Remains The Same (Disc 2)	Stairway To Heaven
The Song Remains The Same (Disc 2)	Moby Dick
The Song Remains The Same (Disc 2)	Whole Lotta Love
Physical Graffiti [Disc 1]	Custard Pie
Physical Graffiti [Disc 1]	The Rover
Physical Graffiti [Disc 1]	In My Time Of Dying
Physical Graffiti [Disc 1]	Houses Of The Holy
Physical Graffiti [Disc 1]	Trampled Under Foot
Physical Graffiti [Disc 1]	Kashmir
Physical Graffiti [Disc 2]	In The Light
Physical Graffiti [Disc 2]	Bron-Yr-Aur
Physical Graffiti [Disc 2]	Down By The Seaside
Physical Graffiti [Disc 2]	Ten Years Gone
Physical Graffiti [Disc 2]	Night Flight
Physical Graffiti [Disc 2]	The Wanton Song
Physical Graffiti [Disc 2]	Boogie With Stu
Physical Graffiti [Disc 2]	Black Country Woman
Physical Graffiti [Disc 2]	Sick Again

Agora conseguimos realizar um filtro feito para álbum e também para nome de música!

Ao longo da aula vimos como refinar uma consulta pronta a partir de novos parâmetros. Também vimos que não é necessário criar uma nova consulta toda vez que queremos incluir um parâmetro distinto!