

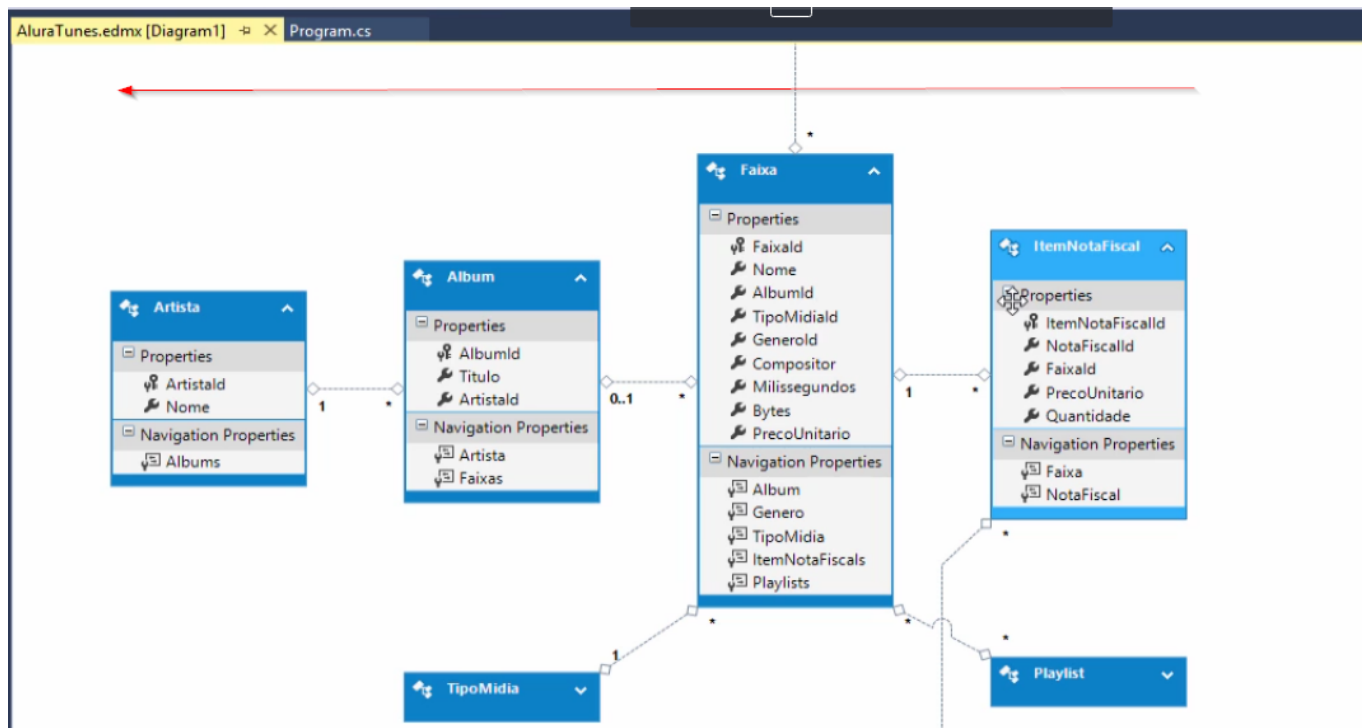
## 2 - Linq to entities Sum

### Transcrição

O novo pedido do cliente é realizar uma consulta que calcule o total de vendas de um determinado artista. Para fazer isso é necessário entender qual é o modelo de vendas adotado pelo cliente!

O modelo de venda inicia por uma nota fiscal que contém diversos itens e cada um deles representa um elemento que está sendo comprado pelo consumidor. Nós vamos filtrar os objetos a partir das notas e também, utilizando os `artistas`. Por exemplo, em uma nota fiscal que contém cinco itens, três podem ser de um mesmo artista e outros dois de artistas diversos.

Vamos visualizar o `edmx` do Entity Framework.



Iniciaremos pelo modelo `ItemNota Fiscal`. Depois, vamos até `Faixa`, e em seguida `Album`. Por fim, `Artista`. Será por este último que faremos o filtro da consulta.

Para começar, abriremos o arquivo `Program.cs`. Vamos iniciar a consulta a partir do item de nota fiscal. Primeiramente utilizaremos a variável `query` e isso deverá ser igual a:

```
from inf in contexto.ItemNotaFiscal
```

Depois, acrescentaremos o `select inf;` e também o `var inf in query`. Vamos colocar logo abaixo o `Console.WriteLine()` e passaremos como parâmetro: `inf.Faixa.Nome`, `inf.Quantidade` e `inf.PrecoUnitario`. Teremos o seguinte:

```
static void Main (string[] args)
{
    using(var contexto = new AluraTunesEntities())
    {
        var query = from inf in contexto.ItemNotaFiscal
                    select inf;
```

```

foreach (var inf in query)
{
    Console.WriteLine(inf.Faixa, inf.Quantidade, inf.PrecoUnitario);
}
}

```

Ao testarmos a aplicação, teremos como resultado todos os itens de notas fiscais. Mas, queremos somente aquilo que mais nos interessa. Por isto, utilizaremos a cláusula:

```

where inf.Faixa.Album.Artista.Nome == "Led Zeppelin"

```

Com as alterações teremos:

```

static void Main (string[] args)
{
    using(var contexto = new AluraTunesEntities())
    {
        var query = from inf in contexto.ItemNotaFiscal
                     where inf.Faixa.Album.Artista.Nome == "Led Zeppelin"
                     select inf;

        foreach (var inf in query)
        {
            Console.WriteLine(inf.Faixa, inf.Quantidade, inf.PrecoUnitario);
        }
    }
}

```

Observe que conseguimos economizar três `join` s ao escrever o `where` . No `Console` falta a formatação, então, escreveremos junto do `Console.WriteLine()` o `"{0}\t{1}\t{2}"` . Teremos:

```

static void Main (string[] args)
{
    using(var contexto = new AluraTunesEntities())
    {
        var query = from inf in contexto.ItemNotaFiscal
                     where inf.Faixa.Album.Artista.Nome == "Led Zeppelin"
                     select inf;

        foreach (var inf in query)
        {
            Console.WriteLine("{0}\t{1}\t{2}"inf.Faixa, inf.Quantidade, inf.PrecoUnitario);
        }
    }
}

```

O que faremos agora é calcular os valores totais de cada um dos itens! Assim, vamos modificar o `select` para trazer o `totalDoItem` e igualaremos a isso uma multiplicação, pois o valor que desejamos não existe pronto. Como temos apenas os valores de `PrecoUnitario` e `Quantidade` , precisaremos multiplicar um pelo outro e desta forma obter o `Total` .

```
totalDoItem = inf.Quantidade * inf.PrecoUnitario
```

Portanto, acrescentaremos um objeto anônimo no `select`, depois, o `new`. Ao mesmo tempo, iremos alterar o `Console.WriteLine()` para que ele indique o `total`. O código ficará assim:

```
static void Main (string[] args)
{
    using(var contexto = new AluraTunesEntities())
    {
        var query = from inf in contexto.ItemNotaFiscal
                     where inf.Faixa.Album.Artista.Nome == "Led Zeppelin"
                     select new { totalDoItem = inf.Quantidade * inf.PrecoUnitario }

        foreach (var inf in query)
        {
            Console.WriteLine("{0}", inf.totalDoItem);
        }

        Console.ReadKey();
    }
}
```

Ao executarmos o código, teremos o total de vendas por item do artista `Led Zeppelin`. Mas, nosso objeto ainda não está completo! Para concluí-lo, deixaremos a seguinte parte comentada:

```
//foreach (var inf in query)
//{
//    Console.WriteLine("{0}", inf.totalDoItem);
//}
```

Abaixo disso, utilizaremos a variável `totalDoArtista` e junto, o método `Sum()`, de soma. Assim, todos os itens que estão no `select` serão somados. Teremos:

```
//foreach (var inf in query)
//{
//    Console.WriteLine("{0}", inf.totalDoItem);
//}

var totalDoArtista = query.Sum()
```

Nosso interesse é acessar o objeto que está sendo retornado ao `select`. Para isto, é preciso adicionar no método `Sum()` uma expressão `lambda` que deve considerar somente a propriedade `totalDoItem`. Portanto, nós vamos passar isso para o `query.Sum()` utilizando:

```
q => q.totalDoItem
```

Vamos imprimir o resultado adicionando a seguinte linha:

```
Console.WriteLine("Total do artista: R$ {0}", totalDoArtista)
```

Após as alterações, teremos:

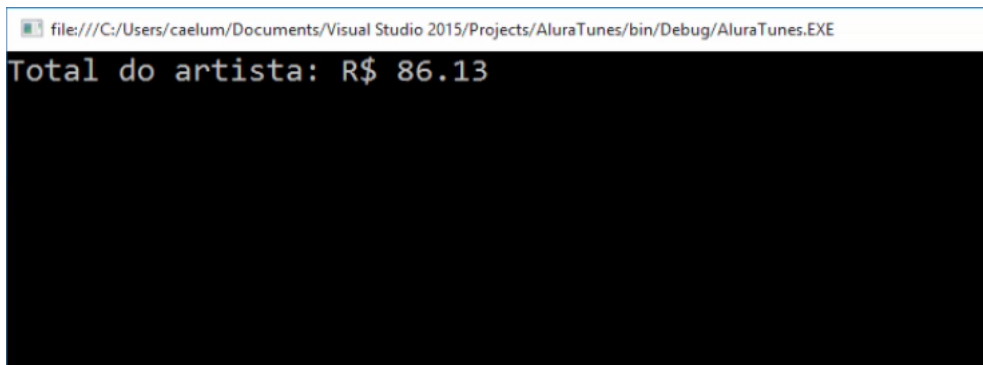
```
//foreach (var inf in query)
//{
//    Console.WriteLine("{0}", inf.totalDoItem);
//}

var totalDoArtista = query.Sum(q => q.totalDoItem);

Console.WriteLine("Total do artista: R$ {0}", totalDoArtista);

Console.ReadKey();
```

Rodando este código temos o seguinte resultado:



Por fim, chegamos ao total de vendas que o artista realizou.