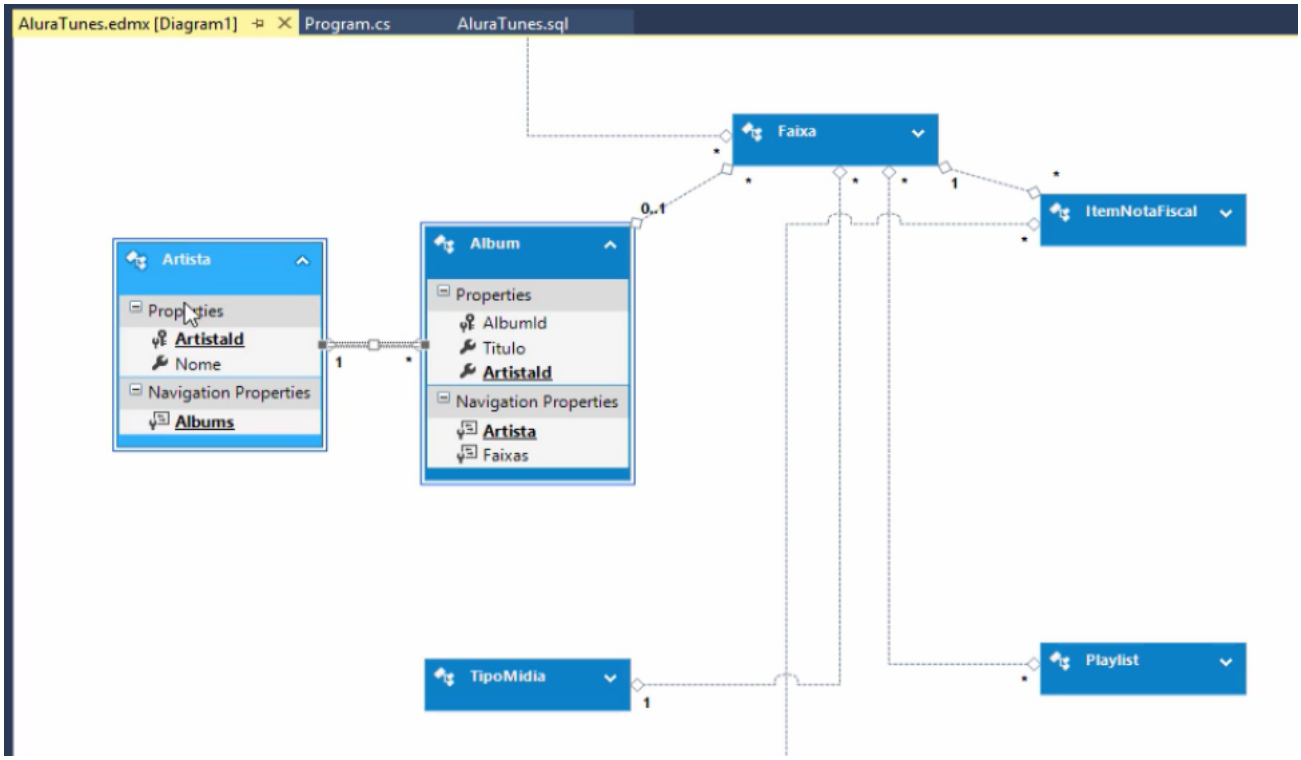
02

2 - Linq to entities sem join

Transcrição

Na última aula, vimos como realizar uma consulta utilizando não apenas o nome, mas também os álbuns. Para realizar a consulta optamos por utilizar o `join` que é ótimo, entretanto, também acaba sendo chato de implementar! Portanto, seria possível obter o mesmo resultado sem o `join` ?

Abrindo o arquivo `AluraTunes.edmx` , poderemos observar as diferentes propriedades que existem. Observe a imagem abaixo:



Note que existe a propriedade `ArtistaId` que é utilizada junto com o `join` e existe também a propriedade `Artista` , nós vamos acessar essa segunda! Para isso, criamos uma nova consulta:

```
var query2 = from alb in contexto.Albums
```

Adicionamos `select alb` e vamos inserir um código que imprima as informações da consulta no `Console` . Portanto, vamos escrever que para cada álbum na `query2` deve ser impresso um novo álbum:

```
foreach(var album in query2)
```

Abaixo disso, escreveremos o `Console.WriteLine(album,Titulo)` e acrescentaremos uma pequena separação com o `Console.WriteLine()` . Teremos:

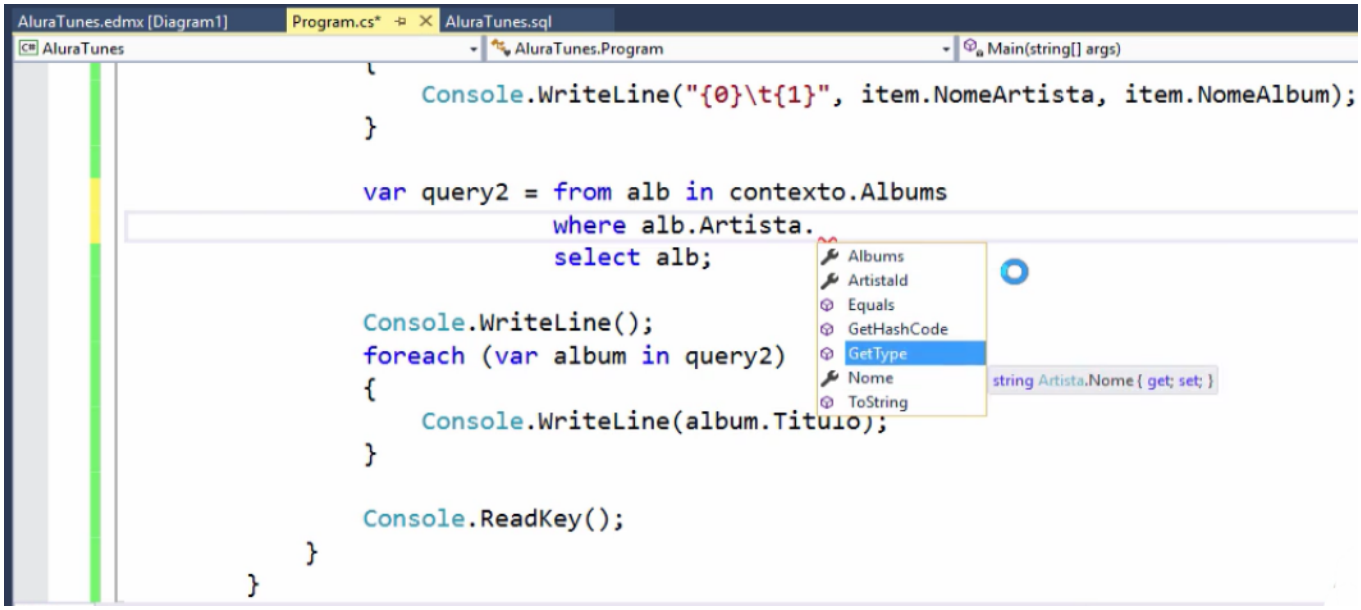
```
var query2 = from alb in contexto.albums
             select alb;

Console.WriteLine()
```

```
foreach(var album in query2)
{
    Console.WriteLine(album,Titulo);
}
```

Ao executarmos o código, temos como resultado todos os itens! Como nosso objetivo é filtrar os resultados vamos adicionar o `where` para acessar a propriedade `Artistas` escrevendo:

```
where alb.Artista.()
```

Ao acrescentarmos `.` depois do `Artista` aparece uma lista com diversas propriedades:



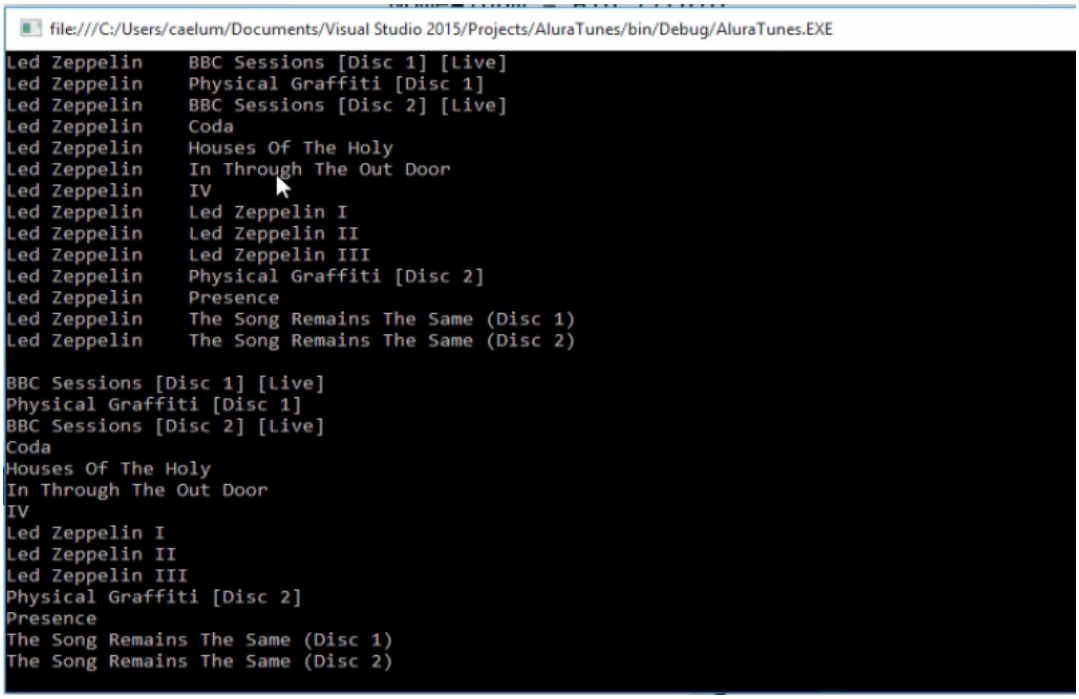
Vamos utilizar a propriedade `Nome` e fazer o filtro a partir disso! Dessa maneira acrescentaremos o `Contains` :

```
where alb.Artista.Nome.Contains()
```

E passamos para isso a variável de busca: `textoBusca` . Teremos:

```
var query2 = from alb in contexto.Albums
             where alb.Artista.Nome.Contains(textoBusca)
             select alb;
```

Se rodamos isso temos o seguinte:



Dessa maneira, o resultado apresentado na consulta que está localizada na parte inferior da imagem é feita sem o `join` e a de cima com o `join` . Na nova consulta sem o `join` ainda faltará acrescentar o "Nome do artista". Portanto, abaixo do `where` iremos inserir `select new` e dentro passamos as propriedade "Nome do artista" e "Albuns":

```
NomeArtista = alb.Artista.Nome
```

E também:

```
NomeAlbum = alb.Titulo
```

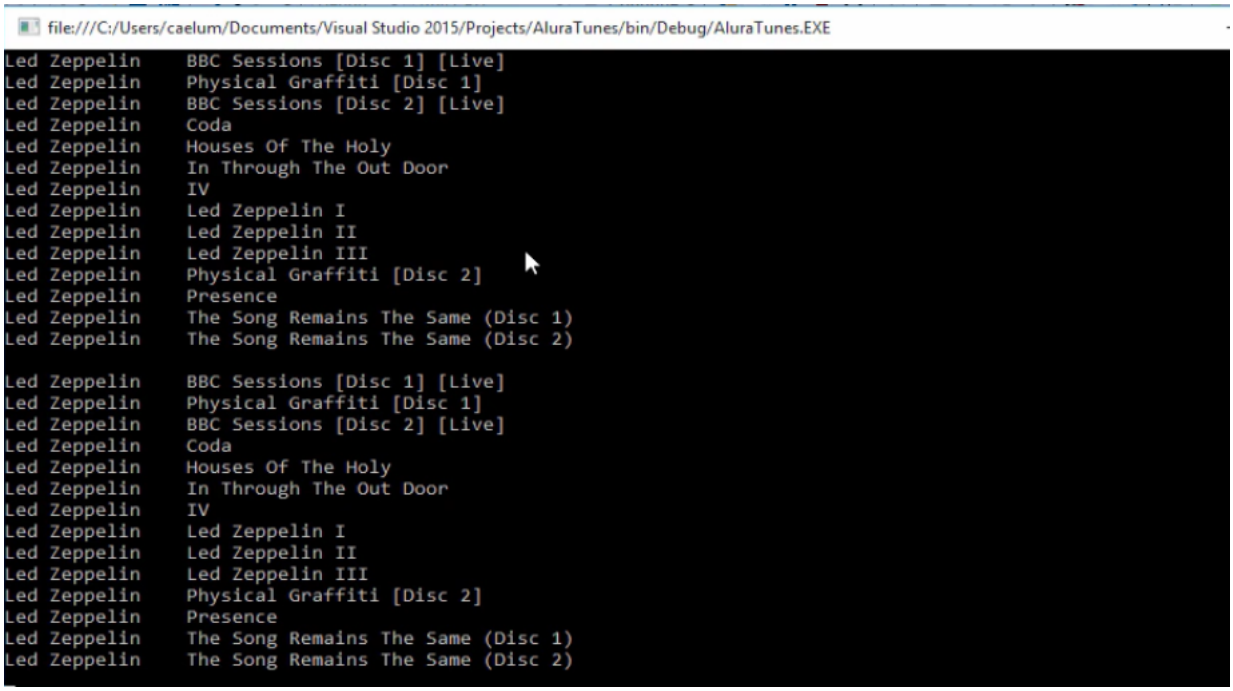
Ainda, vamos adaptar o `Console.WriLine()` para trazer as duas propriedades e também a formatação adequada. Teremos:

```
Console.WriteLine("{0}\t{1}"album.NomeArtista, album.NomeAlbum)
```

Nosso código ficou assim:

```
var query2 = from alb in contexto.Albums
             where alb.Artista.Nome.Contains(textoBusca)
             select new
             {
                 NomeArtista = alb.Artista.Nome,
                 NomeAlbum = alb.titulo
             };

Console.WriteLine();
foreach (var album in query2)
{
    Console.WriteLine("{0}\t{1}"album.NomeArtista, album.NomeAlbum)
}
```

Quando executamos o código, teremos o seguinte resultado :



Note que o resultado da consulta é igual e alcançamos isso sem utilizar o `join` ! Observe também que a consulta sem o `join` fica inclusive mais limpa!

Nesta altura, uma pergunta que pode ocorrer ao aluno é: **Qual a efetiva necessidade de utilizar o `join` ?**

Neste caso específico, o `join` não é necessário. Mas, em alguns momentos é essencial! Por exemplo, no `Entity Framework` , caso ocorra uma consulta na qual todos os álbuns possuem o mesmo nome do artista, não é possível utilizar a propriedade de navegação, portanto, é aconselhável utilizar o `join` !