

3 - Subconsulta

Transcrição

Neste curso trabalharemos com um cliente que possui uma loja on-line. A partir dela ele vai realizar diversos pedidos para que relatórios sejam produzidos.

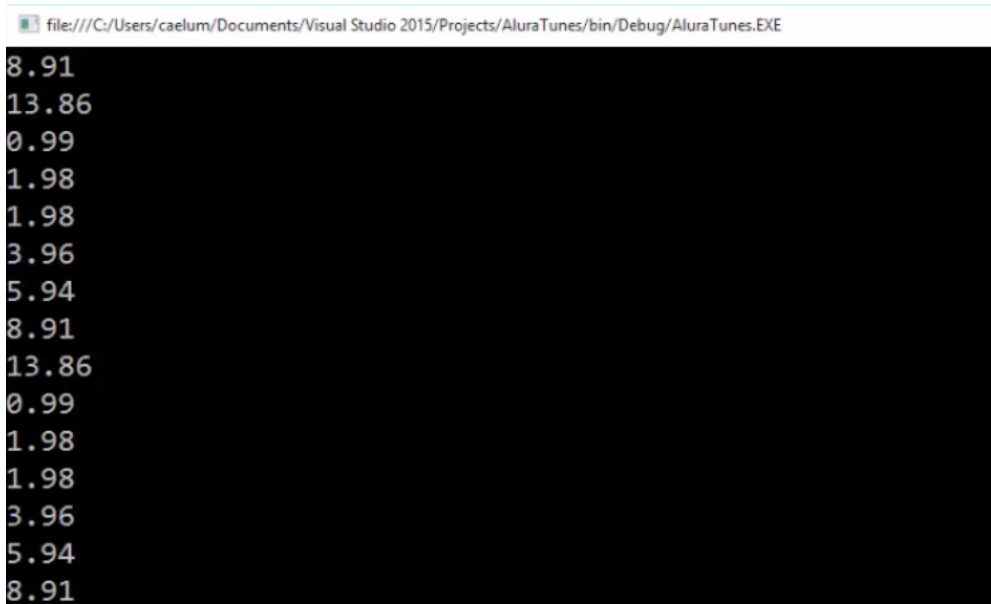
O primeiro pedido do cliente é gerar um relatório que traga as vendas que possuem valor acima da média. Para fazer isso vamos iniciar com uma consulta simples, portanto, vamos introduzir a `var query =` e dentro disso colocamos o `from nf in contexto.NotasFiscais` e o `select nf`. Como é preciso imprimir o conteúdo da consulta, adicionamos o `foreach (var nota in query)` e o `Console.WriteLine(notaFiscal.Total)`. Por fim, acrescentamos o `Console.ReadKey()` para que a janela do resultado não seja imediatamente fechada. Teremos o seguinte código:

```
var query =
from nf in contexto.NotasFiscais
select nf;

foreach (var notaFiscal in query)
{
    Console.WriteLine(notaFiscal.Total);
}

Console.ReadKey();
```

Ao rodar a aplicação teremos uma listagem com os valores totais de notas.



```
file:///C:/Users/caelum/Documents/Visual Studio 2015/Projects/AluraTunes/bin/Debug/AluraTunes.EXE
8.91
13.86
0.99
1.98
1.98
3.96
5.94
8.91
13.86
0.99
1.98
1.98
3.96
5.94
8.91
```

O resultado é uma consulta simples que gera um relatório igualmente simples! A partir de agora vamos incrementá-lo para que traga campos específicos. Para fazer isso modificamos o `select nf` para `select new` e dentro deste último incluímos os dados desejados:

```
Número de nota: Numero = nf.NotaFiscalId Data de nota: Data = nf.DataNotaFiscalId Cliente: Cliente =
nf.Cliente.PrimeiroNome + " " + nf.Cliente.Sobrenome Valor: nf.Total
```


No campo `Cliente` adicionamos o primeiro nome e também o sobrenome, por isso introduzimos `nf.Cliente.PrimeiroNome` e `nf.Cliente.Sobrenome`, respectivamente. O código fica da seguinte maneira:

```
var query =
from nf in contexto.NotasFiscais
select new
{
    Numero = nf.NotaFiscalId,
    Data = nf.DataNotaFiscalId,
    Cliente = nf.Cliente.PrimeiroNome + " " + nf.Cliente.Sobrenome,
    Valor = nf.Total
};
```

Ao incluir estas informações é preciso alterar o relatório para trazer todos os campos descritos acima. Assim, incluímos `notaFiscal.Numero`, `notaFiscal.Data`, `notaFiscal.Cliente` e `notaFiscal.Valor`. Junto disso adicionamos uma string de formatação, a `"{0}\t{1}\t{2}\t{3}"`, que refere-se as quatro colunas da consulta. Nosso código fica da seguinte maneira:

```
foreach (var notaFiscal in query)
{
    Console.WriteLine("{0}\t{1}\t{2}\t{3}",
        notaFiscal.Numero,
        notaFiscal.Data,
        notaFiscal.Cliente,
        notaFiscal.Valor);
}
```

Rodando a aplicação o resultado é um relatório que traz os dados de número, data, cliente e valor:



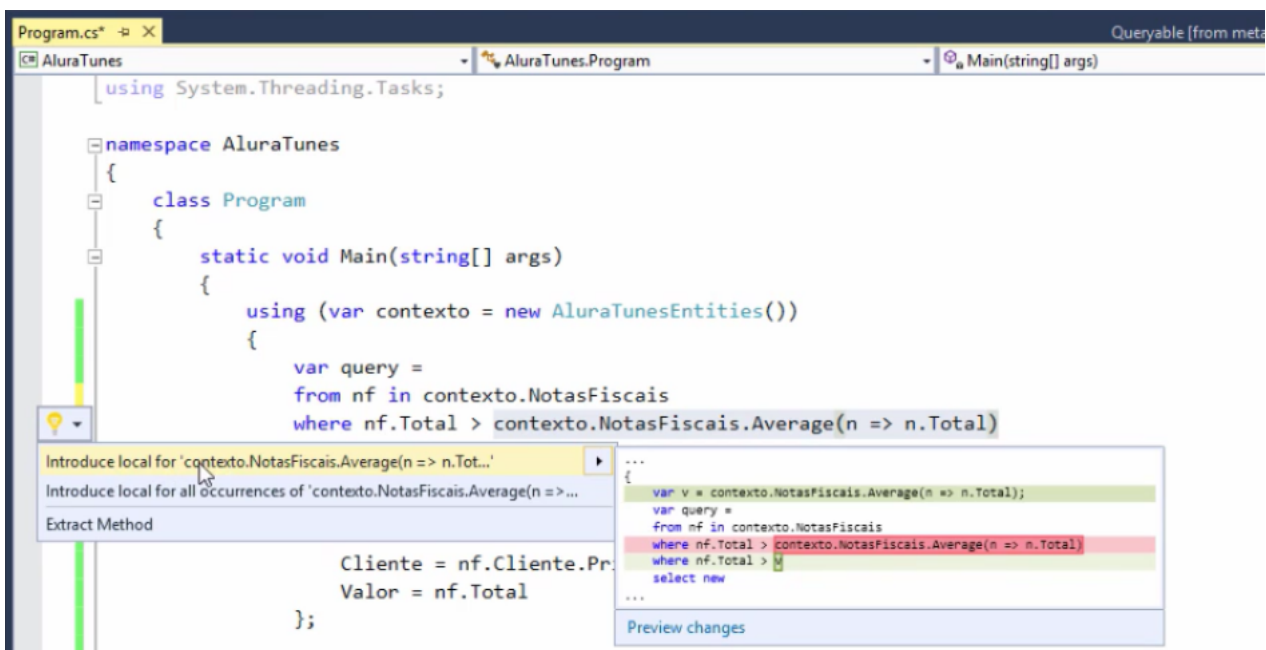
47	7/16/2009 12:00:00 AM	Jennifer Peterson	13.86
48	7/24/2009 12:00:00 AM	Robert Brown	0.99
49	8/6/2009 12:00:00 AM	Edward Francis	1.98
50	8/6/2009 12:00:00 AM	Aaron Mitchell	1.98
51	8/7/2009 12:00:00 AM	Joao Fernandes	3.96
52	8/8/2009 12:00:00 AM	Niklas Schröder	5.94
53	8/11/2009 12:00:00 AM	Terhi Hämäläinen	8.91
54	8/16/2009 12:00:00 AM	Phil Hughes	13.86
55	8/24/2009 12:00:00 AM	Daan Peeters	0.99
56	9/6/2009 12:00:00 AM	Kara Nielsen	1.98
57	9/6/2009 12:00:00 AM	Alexandre Rocha	1.98
58	9/7/2009 12:00:00 AM	Fernanda Ramos	3.96
59	9/8/2009 12:00:00 AM	Jack Smith	5.94
60	9/11/2009 12:00:00 AM	John Gordon	8.91
61	9/16/2009 12:00:00 AM	Aaron Mitchell	13.86
62	9/24/2009 12:00:00 AM	Hugh O'Reilly	0.99
63	10/7/2009 12:00:00 AM	Lucas Mancini	1.98
64	10/7/2009 12:00:00 AM	Stanislaw Wójcik	1.98
65	10/8/2009 12:00:00 AM	Joakim Johansson	3.96

Feito esse procedimento de incluir as diversas informações que nos interessam, vamos seguir com nosso objetivo que é pegar as notas fiscais com valores de venda acima da média. Para encontrar tais itens é preciso criar um filtro, portanto, utilizamos o `where nf.Total`. Para obter a média inserimos um `contexto.NotasFiscais` e junto disso acrescentamos o método `Average()` e passamos para ele a expressão `n -> n.Total`:

```
var query =
from nf in contexto.NotasFiscais
where nf.Total > contexto.NotasFiscais.Average(n => n.Total),
select new
{
    Numero = nf.NotaFiscalId,
    Data = nf.DataNotaFiscalId,
    Cliente = nf.Cliente.PrimeiroNome + " " + nf.Cliente.Sobrenome,
    Valor = nf.Total
};
```

Observe o código acima, ele mostra dois tipos de consulta `query`, uma externa que engloba todo código e outra interna que resume-se a: `contexto.NotasFiscais.Average(n -> n.Total)`. A `query` interna é classificada como `subquery` e essa é uma novidade deste curso! Até o momento não tínhamos nos deparado com isso! Uma dado importante é que as **subqueries podem ser utilizadas em distintos locais da consulta e não apenas no `where`**.

A leitura da consulta, entretanto, está um pouco difícil, pois estamos sobrepondo `query` com `subquery`. Dessa maneira, para facilitar, podemos refatorar a consulta para que seja extraída da definição da subconsulta uma variável. Para fazer isso selecionamos o trecho que corresponde a subconsulta, o `contexto.NotasFiscais.Average(n -> n.Total)`. Aplicamos nele a refatoração através do atalho "`Ctrl + .`" e escolhemos a opção disponível de `Introduce local for 'contexto...'`:



Dessa forma, é criada uma variável chamada "V". Vamos renomeá-la para `queryMedia` e teremos o seguinte:

```
var queryMedia = contexto.NotasFiscais.Average(n => n.Total),

var query =
from nf in contexto.NotasFiscais
where nf.Total > queryMedia
select new
```

```
{
    Numero = nf.NotaFiscalId,
    Data = nf.DataNotaFiscalId,
    Cliente = nf.Cliente.PrimeiroNome + " " + nf.Cliente.Sobrenome,
    Valor = nf.Total
};
```

Ao rodar o código o resultado são valores desordenados. Para organizá-los vamos adicionar o `order by nf.Total` e como queremos que os elementos sejam arrumados de maneira decrescente, adicionamos o `descending`. Temos o seguinte:

```
var query =
from nf in contexto.NotasFiscais
where nf.Total > queryMedia
orderby nf.Total descending
select new
{
    Numero = nf.NotaFiscalId,
    Data = nf.DataNotaFiscalId,
    Cliente = nf.Cliente.PrimeiroNome + " " + nf.Cliente.Sobrenome,
    Valor = nf.Total
};
```

Com esta modificação no código podemos rodá-lo que os objetos serão ordenados de maneira decrescente!

A dúvida é: Estarão todas as notas ordenadas de maneira decrescente ou apenas aquelas acima da média?

Para verificar essa informação, inserimos, abaixo dos campos do `Console`, um comando que indica que a média das notas deve ser impressa. Assim, vamos introduzir o `Console.WriteLine()` e passamos para ele a `"A média é {0}"`. Junto disso colocamos a variável que armazena as expressões, a `queryMedia`. O código final ficará da seguinte maneira:

```
var query =
from nf in contexto.NotasFiscais
where nf.Total > queryMedia
orderby nf.Total descending
select new
{
    Numero = nf.NotaFiscalId,
    Data = nf.DataNotaFiscalId,
    Cliente = nf.Cliente.PrimeiroNome + " " + nf.Cliente.Sobrenome,
    Valor = nf.Total
};

foreach (var notaFiscal in query)
{
    Console.WriteLine("{0}\t{1}\t{2}\t{3}",
        notaFiscal.Numero,
        notaFiscal.Data,
        notaFiscal.Cliente,
        notaFiscal.Valor);
}

Console.WriteLine("A média é {0}", queryMedia);
```

```
Console.ReadKey();
```

Rodando o código o resultado obtido é um relatório que mostra objetos superiores a média, são mostrados os itens acima de 5,65 até o maior valor de venda, o de 25,86 :

```
Select file:///C:/Users/caelum/Documents/Visual Studio 2015/Projects/AluraTunes/bin/Debug/AluraTunes.EXE
404      11/13/2013 12:00:00 AM  Helena Holy      25.86
299      8/5/2012 12:00:00 AM    Richard Cunningham  23.86
194      4/28/2011 12:00:00 AM    Hugh O'Reilly    21.86
96       2/18/2010 12:00:00 AM    Ladislav Kovács  21.86
201      5/29/2011 12:00:00 AM    Victor Stevens   18.86
89       1/18/2010 12:00:00 AM    Astrid Gruber    18.86
88       1/13/2010 12:00:00 AM    Luis Rojas       17.91
306      9/5/2012 12:00:00 AM    Frantisek Wichterlová  16.86
313      10/6/2012 12:00:00 AM    Isabelle Mercier  16.86
208      6/29/2011 12:00:00 AM    Bjorn Hansen     15.86
103      3/21/2010 12:00:00 AM    Frank Ralston    15.86
193      4/23/2011 12:00:00 AM    Fynn Zimmermann  14.91
40       6/15/2009 12:00:00 AM    Hannah Schneider  13.86
355      4/10/2013 12:00:00 AM    Madalena Sampaio  13.86
278      5/4/2012 12:00:00 AM    Edward Francis   13.86
159      11/24/2010 12:00:00 AM    Ellie Sullivan   13.86
257      2/1/2012 12:00:00 AM    Joao Fernandes   13.86
397      10/13/2013 12:00:00 AM    Patrick Gray     13.86
82       12/18/2009 12:00:00 AM    Julia Barnett    13.86
```

Nesta aula aprendemos a utilizar uma subconsulta para deixar o código mais simples e fácil de manter! Caso quiséssemos modificar a consulta mãe teríamos que expandi-la, acrescentar cláusula, etc e isso dificultaria a consulta como um todo. Dessa maneira, aprendemos a separar em queries menores para facilitar nossa vida!